Alma Mater Studiorum – Università di Bologna

DOTTORATO DI RICERCA IN

Diritto e nuove Tecnologie: indirizzo "Informatica Giuridica e Diritto dell'Informatica"

Ciclo XXV

Settore Concorsuale di afferenza: 12/H3 - FILOSOFIA DEL DIRITTO

Settore Scientifico disciplinare: IUS/20 - FILOSOFIA DEL DIRITTO

INTERPRETING JUDGEMENTS USING KNOWLEDGE REPRESENTATION METHODS AND COMPUTATIONAL MODELS OF ARGUMENT

Presentata da: Marcello Ceci (marcello.ceci@gmail.com)

Coordinatore Dottorato	**Relatore**	**Correlatore**
Prof. G. Sartor	Prof.ssa M. Palmirani	Prof. T. F. Gordon

Esame finale anno 2013

Abstract

The goal of the present research is to define a Semantic Web framework for precedent modelling, by using knowledge extracted from text, metadata, and rules, while maintaining a strong text-to-knowledge morphism between legal text and legal concepts, in order to fill the gap between legal document and its semantics. The framework is composed of four different models that make use of standard languages from the Semantic Web stack of technologies: a document metadata structure, modelling the main parts of a judgement, and creating a bridge between a text and its semantic annotations of legal concepts; a legal core ontology, modelling abstract legal concepts and institutions contained in a rule of law; a legal domain ontology, modelling the main legal concepts in a specific domain concerned by case-law; an argumentation system, modelling the structure of argumentation. The input to the framework includes metadata associated with judicial concepts, and an ontology library representing the structure of case-law. The research relies on the previous efforts of the community in the field of legal knowledge representation and rule interchange for applications in the legal domain, in order to apply the theory to a set of real legal documents, stressing the OWL axioms definitions as much as possible in order to enable them to provide a semantically powerful representation of the legal document and a solid ground for an argumentation system using a defeasible subset of predicate logics. It appears that some new features of OWL2 unlock useful reasoning features for legal knowledge, especially if combined with defeasible rules and argumentation schemes. The main task is thus to formalize legal concepts and argumentation patterns contained in a judgement, with the following requirement: to check, validate and reuse the discourse of a judge - and the argumentation he produces - as expressed by the judicial text.

Table of Contents

Introduction

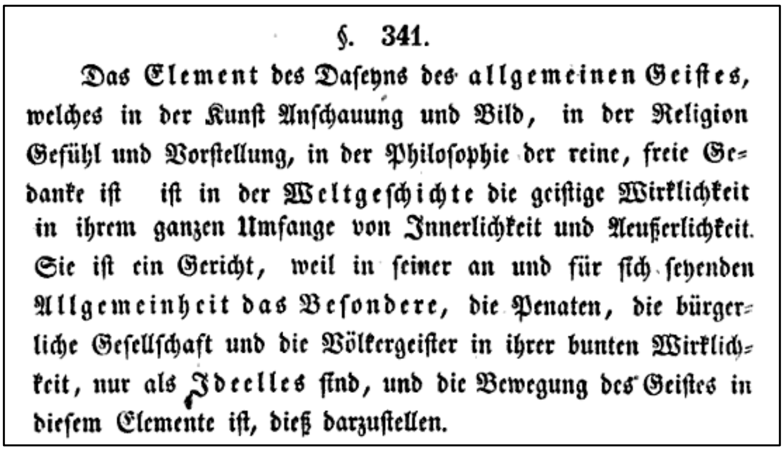
> "The element in which the universal mind exists in art is intuition and imagery, in religion feeling and representative thinking, in philosophy pure freedom of thought. In world history this element is the actuality of mind in its whole compass of internality and externality alike. **World history is a court of judgement because in its absolute universality, the particular** – i.e. the Penates, civil society, and the national minds in their variegated actuality – **is present as only ideal, and the movement of mind in this element is the exhibition of that fact**."
>
> – Georg Wilhelm Friedrich Hegel, *Grundlinien der Philosophie des Rechts*, §341.

Capturing the semantics of human-created texts for processing by machines is not a linear process. In order to provide a comprehensive representation of the contents of a document it is necessary to adopt multiple perspectives, and to give account for different aspects, and depths, of representation.

The paragraph of G. W. F. Hegel cited above is an example of this complexity: different readers may understand different contents from that piece of text. Probably these interpretations have different degrees of accuracy, in the light of external factors (the preceding and following paragraphs of *Der Philosophie des Rechts*, the specific meaning attributed by Hegel to certain terms or combinations of them, and so on), and one may exclude the other, yet it could well be that several meanings are expressed by that combination of words and, in order to *capture the semantics* of that text, all of them must be represented. An *authentic* interpretation (that intended by Hegel himself) could exist, indicated in some other text, but is it sufficient to say that all other possible interpretations are *wrong*, or *meaningless*?

The multiplicity of perspectives does not involve only the translation between signs and concepts, but also the relation between signs located sharing the same source. The source of Hegel's paragraph presented above is reproduced in Figure 0.1, but how many people could say so without looking at the figure's description? The two texts could be bound to the same map of (possible) meanings, but yet those two are distinct in several aspects: the introducing quote is in English language (and *translation* has many entailments), presents itself in the form of a piece of text inside a digital document, and there are additions in the text's format (words highlighted in bold) which are not from the original author. The text in Figure 0.1, instead, is in German language, presents itself in the form of a picture, and reproduces the text format of the original edition of 1821. It is a derivation of the original document, just like the English citation, yet their differences with the original are totally diverse.

Legal documents require further depth for representing its semantics, as they contain not just plain information but rather codify an order of an authority which can be translated into logical operators, but whose syntax is not fixed. Unlike a text from Hegel, where it is reasonable to expect some explanation from the original author (often contained in other works) on the intended meaning of the combination of signs, interpretation of legal documents is a different matter. The language used is important by itself, its conventional meaning being codified by the legal system. However it is known that assigning a meaning to legal dispositions not straightforward: there are gray areas in the interpretation of legal concepts, and the effects of legal acts are susceptible to change in time, either depending on a change of the legal text itself or on external influences (i.e. of other norms, or of judgements). In order to build a model that represents this net of information for the use by Computers, it is necessary to rely on several disciplines:


§. 341.

Das Element des Daseyns des allgemeinen Geistes, welches in der Kunst Anschauung und Bild, in der Religion Gefühl und Vorstellung, in der Philosophie der reine, freie Gedanke ist ist in der Weltgeschichte die geistige Wirklichkeit in ihrem ganzen Umfange von Innerlichkeit und Aeußerlichkeit. Sie ist ein Gericht, weil in seiner an und für sich seyenden Allgemeinheit das Besondere, die Penaten, die bürgerliche Gesellschaft und die Völkergeister in ihrer bunten Wirklichkeit, nur als Ideelles sind, und die Bewegung des Geistes in diesem Elemente ist, dieß darzustellen.

Fig. 0.1 – The original text of the citation in the beginning of this chapter, from [Hegel 1821].

- **Law**, to understand precisely the concepts contained in legal documents and to identify where legal reasoning diverges from other forms of reasoning;
- **Logics**, to model the meaning of the dispositions;
- **Argumentation theory**, to represent the relations between speech acts;
- **Information Technologies (ITs)**, for encoding these information in a machine-readable way.

0.1. Object of the research

Object of the present research is to represent the semantics of judicial documents and to encode such information in a machine-readable form, giving account for the multiple perspectives of a legal document's form and of its contents and always maintaining a strict link between the legal concepts and the piece(s) of text that originate them. In three years of research, a *judicial framework* was realized, composed by a set of computational ontologies written in the Ontology Web Language (OWL) and by a rule base written in different languages (LKIF-Core, Clojure, LegalRuleML).

0.1.1. Judgements

The kind of legal knowledge that the present research tries to model is *judgement*, used to identify three different concepts:

- The action performed by the judge when analyzing a claim (**adjudication**);
- The action performed by the judge when applying laws to facts (**interpretation** or *subsumption*);
- The legal act (document) containing those actions, also called **judicial decision**. The information contained in the legal act is important for two reasons:
    - It contains information on the *legal consequences* applied on the parties of the trial (it is hence a *source of rights* or *obligations*);
    - The legal reasoning contained in it is capable of influencing successive judgements on the same matter (it hence constitutes a *precedent*).

Judgements are core elements of legal knowledge worldwide: by settling conflicts and sanctioning illegal behaviours, judicial activity enforces law provisions within national borders, therefore supporting the validity of laws as well as the sovereignty of the government that issued them. Moreover, precedents (or case-law) are a fundamental source for law interpretation, to the point that the exercise of jurisdiction can even influence the scope of the same norms it has to apply, both in common law and civil law systems – although to different extents. The AI & Law research community has gathered significant results on the representation of precedents since the 1980s, with different approaches: legal case-based reasoning [Ashley 2009, Bruninghaus 2008], ontology-based systems [Mommers 2010], and more recently argumentation [Prakken 2008, Gordon and Walton 2009].

The present research focuses on a key aspect of judgement: the *judicial interpretation*. More precisely, the research will try to provide a complete and functional representation of *judicial subsumptions* acts of judgement that subsume facts under abstract categories. The quote of Hegel in Figure 0.1 can be analysed to get a first insight into the kind of judgement which is the object of the present work. In this paragraph (parts in bold) he *implicitly* defines a *court of judgement* as being an authority that verifies the correspondence between concepts and external reality. It is a perspective on judgement that is particularly helpful for Information Technologies, as it focuses the attention on the concept of logical truth, rather than on the concept of justice. Following Hegel, in fact, "Truth in philosophy means that concept and external reality correspond". This definition of Hegel is particularly interesting when compared to the concept of *judicial subsumption*, which is the main concept modelled by the framework presented by the present research: the concept of judicial subsumption identifies the operation performed by the judge when he considers a material circumstance (a fact, a document, a behaviour) and applies a legal status (a legal qualification or attribute) to it.

This is the meaning intended for *judicial interpretation* or *judicial subsumption* in the present research, whose main task is to correctly *identify* the concept in legal texts, *classify* it in an ontology of legal concepts, *reason* on it with logics, and use it for *argumentation* in legal discussions.

0.2. Contents of the thesis

0.2.1. Judicial reasoning

The thesis begins with a presentation of judicial reasoning-related disciplines. After a brief overview of the main contributors of philosophy of law and legal reasoning, the various families of logics are presented. Such an extensive presentation of logic languages is rendered necessary by the fact that the various applications of ITs and of the Semantic Web rely on different combinations of these languages: it is hence necessary to outline the features that justify one choice for informatics model rather than another. General argumentation theory and diagramming is then presented, followed by a review of the most significant applications in the field of Artificial Intelligence (AI). Finally, the state-of-the-art of Artificial Intelligence and Law (AI&Law) studies related to judicial reasoning is presented.

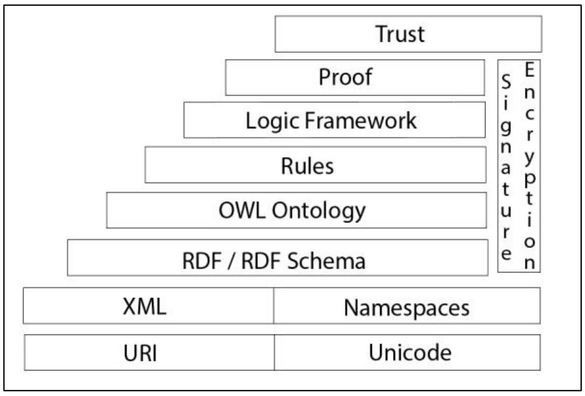


Fig. 0.2 – The Semantic Web stack of technologies.

0.2.2. The Semantic Web and AI & Law

The second chapter is still devoted to a state-of-the-art presentation, this time on the stack of technologies which constitutes the Semantic Web – an offspring of the World Wide Web where data are interconnected in a semantically aware net of relations. Semantic Web-aware technologies can exploit this net of data, automatically retrieving structured information from the Internet and processing it to provide advanced reasoning services on an immense Knowledge Base (KB). The layers of the Semantic Web stack, represented in Figure 0.2 (from the bottom to the top), are all presented in the chapter, together with their main applications in the legal domain:

– **Identifiers and character set**, providing a "name" for entities on the web and a set of character to spell those names and communicate information about their content;

– **Syntax**, setting the pattern for the additional description of the entities on the web;

– **Data interchange**, allowing to establish relations among different sets of entities on the web;

– **Ontologies**, introducing restrictions to these relations in order for the knowledge base to be valid (free from errors) and consistent (free from contradictions).

– **Unifying logic**, performing complex reasoning on the knowledge base which involves different sets of entities on the web;

– **Proof and trust**, ensuring that the knowledge base and the reasoning performed on it are authored, substantially correct, and up-to-date.

0.2.3. The judicial framework: ontology

The third chapter contains a presentation of research project, and a review of its first steps. Relying on existing technologies for document markup, the project built a *set of computational ontologies* (a *core ontology* providing the taxonomy of a judicial decision's structure and of its main components, and a *domain ontology* containing knowledge on the domain's facts, legal statuses and norms) upon the following tasks:

- Semantic representation of the information contained in the decision's text;
- Enrichment of the information through identification and classification of legal, material, and judicial instances;
- Highlighting of a legal norm's applicability to a material instance and the relevancy of a material instance to a legal norm, with an open world perspective;
- Inferencing new possible applicabilities of legal statuses coming from norms or precedents, in an open world perspective.

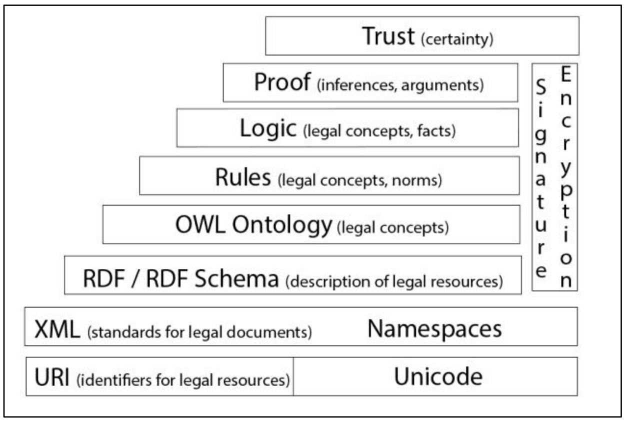


Fig. 0.3 - Tim Berners Lee's Semantic Web layer cake, adapted to the legal domain in [Sartor 2009]

The last two tasks, however, could not be fully achieved, because of the necessity of balancing between a faithful representation of the source legal document and the syntactic expressivity in the model of legal semantics. In order to set up the knowledge base for argumentation, the axioms covering these two tasks had in fact to be translated into a rule base in the upper layer of the Semantic Web stack of technologies.

0.2.4. The judicial framework: rules and argumentation

The trade-off between an efficient and versatile representation of the decision's text and an advanced reasoning on the legal contents suggested a division of the two aspects, creating ad argumentation layer on top of the core and domain ontologies. The set of ontologies can be fully oriented at the representation of the decision's text and at the enrichment of some of the text's semantic contents, while the argumentation on the applicability of norms or judicial precedents is carried out by a different tool which imports the knowledge contained in the set of ontologies. The two elements constitute a *judicial framework*, useful for filling the gap between the original legal document and the data representing its contents[1].

[1] The judicial framework presented in Chapter 3 and Chapter 4 was shown in [Ceci 2012] at the 2012 RuleML International Doctoral Consortium of the ECAI Conference in Montpellier, and the paper received the Best Paper Award within the Doctoral Consortium.

In order to build an effective logics- and rule-based argumentation layer, it was necessary to stray the attention away from open-world reasoning and descriptive logics, delving into defeasible logics. This was the purpose of the research periods spent at the Fraunhofer-FOKUS Institute of Berlin, under the tutorship of prof. Thomas Gordon. The research highlighted that the Carneades, in its last implementation of a *Policy Modeling tool*, includes many features which are fundamental to the management of legal cases, in particular the argumentation theory and the defeasible logics, but lacks in representing temporal dimension and in keeping isomorphism with the source document.

An important distinction has to be made on the kind of knowledge being modeled in the fourth chapter: the argumentation reproduced in the framework is not the court trial itself (the sequence of claims and counterclaims brought forward by the parties before a judge), but rather the argumentation contained in the decision's groundings (the part of the judicial decision which contains the motivation of the final adjudication of the claim).

0.2.5. Advanced considerations

The fifth chapter tries to make a sum of the critical aspects met in the preceding chapters, and to elaborate them in order to lay down a set of requirements for the ideal legal rules language, analyzing different approaches to the representation of judicial knowledge: from general data, through legal concepts, up to case-law. Successively, a summary of the requirements for a rule markup language, as identified during the present research, is enumerated. It is divided into two sets of requirements: ontological requirements, for the correct binding of the abstract concepts to the part of the original text, and syntactical requirements, for the correct representation of the connections between abstract legal concepts. Finally, the LegalRuleML language is presented. This language is a work-in-progress, would-be standard language for legal rules. It contains most of the features discussed in the present chapter, and therefore it is taken as the starting point for the ideal legal rules language. The presentation also highlights where the standard built so far fails to capture aspects and semantics of judicial documents, as identified by the present research.

0.2.6. Conclusions: Hybrid engines

In the concluding part of the work, two projects in the field are analyzed in the light of the requirements laid down in chapter 5. Those projects rely on hybrid reasoning, an interesting approach which mixes case-based and rule-based reasoning to capture that kind of composite deduction process that takes place in several branches of human knowledge, from medicine to law. The research thus concludes by suggesting further exploitation of these hybrid engines, in order to verify which of the requirements for legal reasoning can be fulfilled by an extension of these projects. The requirements set by Chapter 5 constitute, in this perspective, a step towards the definition of a standard in the logic and proof layers of the Semantic Web layer cake in the legal domain, as identified in Figure 0.3.

Chapter 1

Judicial Reasoning: a State of the Art.

"Whatever is referred to must exist. Let us call this the axiom of existence."

– John Searle, *Speech Acts: An Essay in the Philosophy of Language*.

"«It is not necessary to accept everything as true, one must only accept it as necessary.» «A melancholy conclusion, » said K. «It turns lying into a universal principle. »"

– Franz Kafka, *The Trial*.

1.1. A brief history of legal reasoning through its main characters

1.1.1. G. W. Leibniz: understanding law through logics

The work of the great mathematician and philosopher Gottfried Wilhelm Leibniz encompasses the full range of human knowledge of the XVII Century. Information Technologies have, however, the luck and the pride of taking foundation in a main part of Leibniz's work. His main contribution to logics is directed by his search for a representation of human matters through arithmetical formula. Working on a purely philosophical perspective, Leibniz evoked the scene of two disputing philosophers which finally sit down at their abaci and – inviting each other with a friendly "*calculemus*!" – begin calculating "the Truth". According to the philosophy of Leibniz, human reasoning could be transformed into a kind of arithmetic using numbers instead of notions. In such a way, the logical truth of a proposition corresponds to the arithmetical truth of a calculation. In Leibniz's example: if the number of animal were 2 and that of rational were 3, then the number of man (the rational animal) would be obtained by the multiplication 3·2. Then the answer to the question 'Is every man a rational being?' could be reduced to the fact that 6 is divisible by 3.

In order to achieve this objective, Leibniz examined all aspects of mathematics: arithmethics, trying to translate Aristotelian Syllogistic (*Calculi universalis elementa* – in [Leibniz 1694], VI, II, 205–216), algebra, looking for the abstract representation of the logical laws (*De formis syllogismorum mathematice definiendis* – in [Couturat 1903], 410–416), and geometry, reproducing the Aristotelian theory on how material things are formed from combinations (in [Leibniz 1666]) and drawing representations of notions upon the Euler and Venn[2] diagrams (*De formae logicae comporatione per linearum ductus* - in [Couturat 1903], 292–321).

Leibniz's work constitutes the basis of the modern mathematics. On one side, it is undeniable that its fundaments are now known as inappropriate for the task: for example, all Leibniz's logic operates with atoms of the form '*A est B*' and therefore is limited to syllogistic logic or monadic predicate calculus, both decidable while, on the contrary, nowadays it is known that the predicate calculus with dyadic (*A*+*B*) relations is not decidable [Davis 1958]. On the other side, however, Leibniz's work is the starting point of a certain way of thinking logics towards the sake of justice.

1.1.2. G. W. F. Hegel: a legal ontology

Georg Wilhelm Friedrich Hegel, great German idealist whose work prosecuted the long road which was already traveled by Plato and Kant, contributes to the construction of the legal ontology through his work *Elements of the Philosophy of Right* (*Grundlinien der Philosophie des Rechts*, [Hegel 1820]). In this book, he defines the main concepts of law (the concepts of right, will, law, morale, ethics, necessity, obligation, property, State..) with a perspective strongly focused on a series of universal principles (some kind of dialectic axioms) from which he derives the definition of various legal concepts. In this short presentation it is not the case to delve into the details of Hegel's thought: it is just important to mention that a large part of Western culture has tried (since ancient times) to build an ontology of all the concepts in reality based on universal values rather than on material phenomena. Such an approach may be considered, in our times, a typical "philosophical" approach in the degenerate meaning of something abstract, verbose, overthought and in the end useless. Of course such an approach is (and can) not give account for the infinite variations of material phenomena, nevertheless it is an approach which directs the thoughts of us people belonging to the Western culture, and which is also reflected in legal sources: statutory law itself, as well as a relevant part of legal doctrine, are in fact based on abstract concepts, the first disposing abstract and general provisions and the second (trying to) provide an explaination for these provisions, often linking them to more abstract – rather than more concrete – mechanics.

1.1.3. J. Bentham: introducing deontic logics

Born in the changing times of the late XVIII Century, Jeremy Bentham was a British philosopher and jurist. As part of his work, he investigated the nature of law and of its essence, which he defined (in *Of Laws In General*) as:

> "[…] an assemblage of signs declarative of a volition conceived or adopted by the sovereign in a state, concerning the conduct to be observed in a certain case by a certain person or class of persons, who in the case in question are or are supposed to be subject to his power: such volition trusting for its accomplishment to the expectation of certain events which it is intended such declaration should upon occasion be a means of bringing to pass,

[2] See 1.3.3. for an insight into logical diagramming.

> and the prospect of which it is intended should act as a motive upon whose conduct is in question.” [Hart 1970]

This definition, which opens his book *On Laws in General*, can be rephrased as “Law is an expression of sovereign’s will concerning the conduct of its subjects and relying on punishment and motivation for its enactment”. Bentham supported utilitarianism: he believed that the actions of men are led by the desire to maximize utility, which means increasing happiness and reducing suffering. Law has the role of leading those actions: it is not the simple expression of the sovereign's preferences concerning the conduct of his subjects, it is rather the sign of his positive intention to influence their conduct. Bentham defines the “force” of a law as the “motives it relies upon for enabling it to produce the effects it aims at.” Bentham therefore conceives law as a system of social control based on motivation. In these concepts lies the origin of deontic logic, a concept that would have been created only fifty years later by Von Wright but which was already conceived by Bentham as a particular logic, different from an Aristotelian logic of understanding. He called it the logic of imperation, or logic of the will. See 1.2.4.1. for a presentation of that logics.

Bentham also imagined a useful instrument, which he called “Pannomium”, providing to the legislator a complete and coherent system of laws. By measuring the force of a law, the legislator can use such a tool to ensure that the addressed agents will follow his wishes as expressed by law. Here Bentham introduces the idea of managing law through logical tools, and foresees that the main issue of that invention would be the practical reasoning on the side of the governed, which can be resumed in the question “which is the valid logic formula for a given law: that in the mind of the legislator, or those understood by the law’s addressees?” The research of Bentham delves deeper into these concepts (and out of the scope of this brief presentation) by discussing of Legal Practical Reasoning.

1.1.4. F. C. von Savigny: the completeness of law

Friedrich Carl von Savigny was a jurist and historian. He was one of the founders of the famous *German Historical School* (*Historische Rechtsschule*), which supported codifications and regarded law as the expression of the convictions of the people, in the same manner as language, customs and practices are expressions of the people. In his view, the law is grounded in a form of popular consciousness called the *Volksgeist*. Von Savigny founded the *Jurisprudence of Concepts* (*Begriffsjurisprudenz*), the first sub-school of legal positivism, according to which the written law must reflect concepts, when interpreted. In general, *Begriffsjurisprudenz* is assigned to three elementary positions, related to each other:

- The given law contains no gaps;
- The given law can be traced back to a logically organized system of concepts (the “pyramid of concepts”);
- New law can be logically deduced from superordinate legal concepts, which themselves are found inductively (“method of inversion”).

The charges against these positions include epistemological and logical naiveté. Begriffsjurisprudenz influenced 19th and 20th century legal scholarship, and, despite not being recognized as a valid model to manage the production of new law (due to alleged flaws in the approach such as obfuscation of values, remoteness from real life situations, a lack of consideration of super-positive law and generally an overestimation of the purely dogmatic method), it still constitutes an important tool to understand, categorize and connect different legal provisions. Jurisprudence, in particular, often give account for such generalizations when trying to subsume heterogeneous norms under the same legal principles.

1.1.5. E. Ehrlich: enriching the legal painting

Among the main critics of the *Begriffsjurisprudenz* is Eugen Ehrlich, an Austrian legal scholar and sociologist whose positions have given birth to the "Living Law School" (*Freirechtsschule*). Influenced by the cultural mood of the times (among other by the work of the Grimm brothers on popular stories and traditions), Ehrlich noted that legal theories recognized law only as a sum of statutes, and judgements gave an inadequate view of the legal reality of a community. He therefore drew a distinction between *norms of decision* and *norms of conduct* (also called social norms). It is the latter that actually governs the life of a society and can justifiably be regarded in popular consciousness (if not necessarily by lawyers) as law. Ehrlich gives the example of customary law and commercial usage, which may develop and be recognized by courts of law as having normative force and legal significance. The point Ehrlich tried to make was that the "living law" which regulates social life may be quite different from the norms for decision applied by courts, and may sometimes attract far greater cultural authority which a legal system cannot safely ignore. In contrast to von Savigny and his *Begriffsjurisprudenz,* Eugen Ehrlich provides a perspective of law "from below", which means that in order to represent the law it is necessary to give account for its thousands of applications. Together, the two approaches provide a first draft for a model of legal knowledge: it is possible to describe the legal system from the perspective of abstract concepts of from single occurrences, but in order to achieve a complete representation neither of the two can be rejected: a complete model of legal knowledge must thus include both approaches, or at least deal with them.

1.1.6. H. Kelsen: describing law as it is

Hans Kelsen was a jurist and a legal philosopher. He was one of the main supporters of *legal positivism*, a general and descriptive theory of law developed by John Austin in opposition to *natural law theory*[3]. Underlying legal positivism is *meta-ethical noncognitivism*, according to which moral claims have no cognitive meaning, and is therefore important to distinguish between moral rules and "positive law", which is the only law concerned by legal studies and corresponds to nothing more than a "specific social technique of a coercitive order" ([Kelsen 1945]). Legal positivism is a sort of meta-theory, (a theory about theories) of law, setting down the requirements that an adequate theory of law must meet.

---

[3] Natural law theory asserts that there is a conceptual connection between law and morality, and that moral values and standards exist independently of people's beliefs and attitudes. According to natural law theory, the moral authority of law is part of the concept of law, and an unjust law cannot be legally valid, since it cannot be a law at all (*lex injusta non est lex*). Sofocles' "*Antigone*" perfectly explains the ancient, never-ending clash of the two views on gods and men.

In his masterpiece Pure Theory of Law (*Reine Rechtslehre*) Kelsen was interested in understanding law as it is, not as it ought to be. His method is structural analysis: more specifically, he provides a set of fundamental legal concepts (legal system, norm, right, duty, sanction, imputation) in order to describe the law in a scientific manner. Pure Theory of Law has the purpose to lay down the theoretical basis for the various disciplines of law (law of contracts, tort law, constitutional law, comparative law…). Its basic methodological aim is to free the study of law from all foreign elements, to avoid methodological syncretism (see [Paulson 1990]).

Kelsen conceives law as a system of norms structured in a *Stufenbau*, a layered stack of norms where norms on a higher level authorize the creation of norms on a lower level. Kelsen gives the maximum importance to the concept of validity: to say that a legal norm is valid, he explains, is to say that it exists, and to say that it exists is to say that it ought to be obeyed or applied, that it has binding force. He, however, accepts as fundamental the distinction between what is (*Sein*) and what is ought to be (*Sollen*), where the latter is to be intended in "the specific sense in which human behavior is determined by a norm" ([Kelsen 1945], p. 37).

Tracing the validity of a given legal norm through the upper layers of the legal system, one finally arrives at the basic norm (*Grundnorm*), described as "the first postulate, upon which the validity of all the norms of our legal system depends" ([Kelsen 1945] p 115). In the opinion of Hägerström reported in [Spaak 2012], Kelsen had difficulties finding the fundament of validity for the *Grundnorm*, nevertheless this structure of the legal system (where validity descends from other norms and not from abstract moral values) and the approach to the *Pure Theory* of law (distinguishing it from external notions) constitute a fundamental contribution to the modelling of legal knowledge. Hans Kelsen not only shaped the positivist approach to legal disciplines, but also provided it with the tools necessary to describe it without necessarily referring to social, economic or physical effects: he defined what part of the human knowledge, reasoning, and behavior pertains to the legal field.

1.1.7. H. L. A. Hart: the law made of rules

Legal positivism was further developed during the last century. Herbert Lionel Adolphus Hart, an influential legal philosopher, while still conceiving law as a system of norms taking foundation from a single, fundamental norm, criticized some aspects of legal positivism. In particular, in *The Concept of Law* ([Hart 1961]) Hart maintains that what is missing in Austin's theory is the idea of a rule. According to Hart, this idea is necessary in order to analyze the concept of a legal obligation. Hart identifies two kinds of rules:

- **Duty-imposing rules**: to say that someone has a (legal or moral) obligation to a certain behavior is to assume a background of rules that makes that behavior standard, and to apply a rule to that person and his behavior. Such duty-imposing rules, [Hart 1961] explains, are "conceived as binding independently of the consent of the individual bound";

- **Rules of change, adjudication and recognition**: they are used to identify, create, modify and extinguish primary rules, and to set up institutions that apply those rules. More specifically:

- *rules of change* give legal power to persons in order for them to modify the legal system;
- *rules of adjudication* constitute organs to apply the law;
- the *rule of recognition* sets the criteria for identifying the rules of the legal system by:
    - identifying and ranking the sources of law;
    - laying down the criteria for validity of a law;
    - imposing a legal duty on legal officials to apply all and only the norms that fulfill the criteria of validity.

It is this distinction, in Hart's words, that constituted "the step from the pre-legal system to a legal world" ([Hart 1961], 91). Such categorizations constitute the ideal starting point for modeling the law as a consistent organism standing on solid foundations. On the same time, however, the law cannot be taken as a perfectly logical system originated exclusively out of deduction from general principles: when facing the reality, norms (and their underlying logics) need always to change their formulas so that they can cope with the material instances. It is Hart himself that warns about that:

> "There can be no final and exhaustive definitions of concepts, even in science.... We can only redefine and refine our concepts to meet the new situations when they arise" [Hart 1983]

1.1.8. W. N. Hohfeld: disambiguating legal language

An important contribution to legal reasoning was brought by the American jurist Wesley Newcomb Hohfeld, whose few law journal articles were collected after his death in [Hohfeld and Cook 1919], constitutes an important step towards understanding the nature of rights and the implications of liberty. In his work as a jurist, Hohfeld noticed that respected colleagues mix various meanings of the term *right*, sometimes using the same word several times, for different meanings, in the same sentence. In his view, such imprecision of language indicated an imprecision of thought, and thus also of the resulting legal conclusions. In order to facilitate reasoning and clarify rulings, Hohfeld attempted to disambiguate the term *right* by breaking it into eight distinct concepts. He defined these terms on the basis of their relationsips, grouping them into four pairs of Jural Opposites and four pairs of Jural Correlatives (see Figure 1.1). Note that this use of the words *right* and *privilege* corresponds respectively to the concepts of *claim rights* and *liberty rights* (see [McCormack 2005]).

Jural Opposites	**Jural Correlatives**
Right / No-right	Right / Duty
Privilege / Duty	Privilege / No-right
Power / Disability	Power / Liability

Immunity / Liability	Immunity / Disability

Fig. 1.1 – Jural Opposites and Jural Correlatives from [Hohfeld, 1919].

If two concepts are correlative, one must always be matched by the other. If A has a *right* against B, this is equivalent to B having a *duty* towards A. If B has *no duty*, which means that B has a *privilege*: he/she can do whatever he or she pleases, having no duty to refrain from doing it, and A has *no right to prohibit* B from doing so. This approach to the classification of legal concepts constitutes an interesting basis for automatic inferencing on the legal bonds which exist between subjects.

Moreover, in Hohfeld's conception each individual is located within a matrix of relationships with other individuals. By summing the rights held and duties owed across all these relationships, the legal expert can identify the degree of liberty (an individual would be considered to have perfect liberty if it is shown that no-one has a right to prevent the given act) and whether the concept of liberty is comprised by commonly followed practices, thereby establishing general moral principles and civil rights.

1.1.9. J. Searle: doing things with words

John Searle, an American philosopher, brought very interesting contribution in the conception and ontology of legal rules through the *speech acts theory*. Together with J. L. Austin, he introduced the concept of *performative language*, a kind of verbal behaviour where not only speaking, but also an act of some sort is involved (i.e. a promise, a curse, a sentence). Performative utterances not only passively describe a given reality, but they directly change the (social) reality they are describing. Searle also defined as declarations those speech acts whose successful performance is sufficient to bring about "the fit between words and world, to make the propositional content true" ([Searle 1975]). He calls this direction of fit *doubled direction*, world-to-word-to-world. Searle is also important for having distinguished between *regulative* and *constitutive rules*: while regulative rules encode a form of behavior which exists before or independently from them, constitutive rules do not regulate but rather create or define new forms of behavior. Rules of a game (i.e. chess) are not limited to regulate the game, but create the whole new possibility of playing that game.

1.1.10. Recent contributions by Italian legal philosophy scholars

In the last fifty years the various tools for the analysis of legal phenomena were further expanded in various directions. Amedeo Giovanni Conte, legal philosopher, delved deep into the study of deontic and legal language. He structured his work (and his representation of normative utterances) in a strongly formal way, and his research is interesting in the perspective of legal language formalization: he studies the same legal provision (or logical statement) in different languages in order to highlight the several hidden implications of translating legal terms. He studied, among else, the different meanings of the term *truth* producing the table in Figure 1.2.

His work represents a precious example of the approach to be kept when trying to formalize a discipline based on language, such as law: the meaning has to be kept distinct from the signs used to express it, and yet the connection between the two has to be ensured. No matter how complex or subtle a distinction in meaning (or notation) may be: it has to be recognized, dissected and represented in the abstract model. *Dura lex, sed lex*.

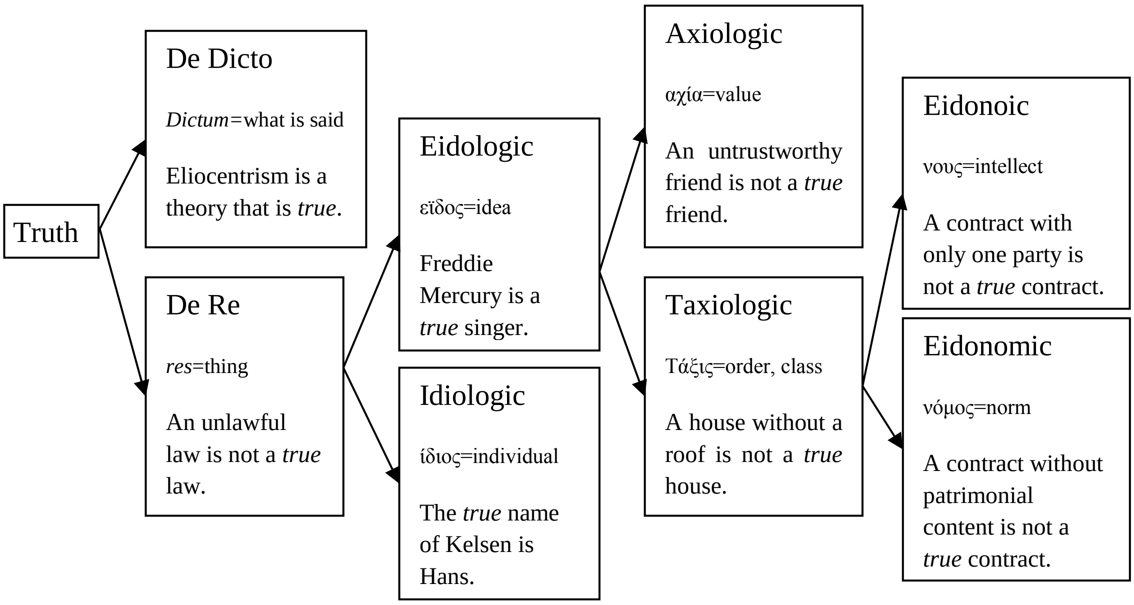


Fig. 1.2 – The "table of thruths" by A. G. Conte. Note that the terminal concepts are five: truth *de dicto* (addressing the content of its object), *idiologic truth* (addressing the object as a single entity), *axiologic truth* (relying on moral evaluations on the object), *eidonoic truth* (relying on human reasoning) and *eidonomic truth* (relying on valid laws).

Enrico Pattaro connected his research for the source of a norm's validity in legal philosophy to Scandinavian legal realism, a very influent school of thought of the last century its main exponent is Alf Ross, which declared the reality of legal system as an undistinguished mix of abstract principles and practical applications, brought together by the *social belief* of the validity of some (and not other) norms. Pattaro develops these positions, investigating how belief and compliance interact in a person's position towards the legal system. He deeply relies on J. Searle and distinguishes between abstract law provisions (*fact-types*) and material occurrences (*fact-tokens*).

These studies bring indeed contributions to the representation of the legal system in formal terms, but nevertheless they astray a bit from the pure concepts of law to get involved with a social or literary analysis of its effects. The work of Giovanni Sartor brings the attention back on the form of the normative principle, but this time with an explicit focus on formal logics and argumentation. Among the most important contributions to the recently growing discipline called Artificial Intelligence & Law, [Sartor 2005] (see 1.4.) presents a comprehensive view on the cognitive approach to law, which draws from formal logics (see 1.2.8.1. for his analysis on the relationships between logics and legal reasoning), argumentation theory and philosophy of law to build a model of legal phenomena that is formally valid, and at the same time consistent with the tradition of philosophy of law. In particular, Sartor affirms the need for defeasible logics to capture the dynamics of legal phenomena: as shown in [Sartor 2012], it is impossible to capture legal norms through what Sartor calls *conclusive reasoning*, where one can always adopt R's conclusions while endorsing R's premises (and one should never reject R's conclusions while endorsing R's premises). It is therefore necessary to adopt a defeasible reasoning approach, where one should, under certain conditions, refrain from adopting the conclusions of an argument though endorsing its premises.

In order to understand the state-of-the-art of legal knowledge representation, it is then necessary to build on different subjects: logics, argumentation theory, and information technologies. In the following sections these subjects are presented, providing the basis for an application of legal theories (among which are those presented in the chapter part about to end) which fully exploits technologies, relies on solid formal roots and is as well consistent with the legal tradition. It is something that can be – as it has, in some sense, already been - achieved.

1.2. Logics

1.2.1. Overview

From an epistemic point of view, a logical argument is a set of one or more propositions (called "premises") that provide inferential warrant for another set of one or more propositions (called "conclusions") To say that one proposition "*x*" (premise) provides inferential warrant for another proposition "*h*" (conclusion), is to say that, according to the argument presented, the truth of *x* would to some extent support the claim that *h* is true. For example, the two premises provide inferential warrant for the conclusion in this classic argument from Aristotle's syllogisms, which corresponds to the *modus ponens* in modern argumentation theory:

All men are mortal.

Socrates is a man.

therefore

Socrates is mortal.

Logic is the study of the different modes of logical inference that different kinds of arguments display.

1.2.2. First-order classical logic

First-order classical logic was established by mathematicians such as George Boole, Gottlob Frege, and Bertrand Russell (see [Boole 1847, Frege 1970, and Russell 1903]) , between the end of the 19th century and the beginning of the 20th, though the first book distinguishing first order logic from other logical formalisms was probably [Hilbert and Ackermann 1927]. The power and the limitations of this formalism were properly circumscribed in the first half of the 20th century, by the work of other great mathematicians, such as [Gödel 1929] and [Tarski 1941].

1.2.2.1. Propositional Logic

Propositional logic is built out of two fundamental elements:

- **atomic propositions** are sequences of words constituting a unit of meaning, each represented by a different atomic symbol (a different propositional constant), and represent the building blocks of logic language;

- **propositional connectives** specify relationships between propositions, with the help of parentheses delimiting their scope.

These connectives express the structure of knowledge supporting the inferences of propositional logic. In the following list the basic logical connectives, which express (one particular interpretation) of the corresponding expression in natural language, are introduced:

Logical connective	English conjunction
$\wedge$	and
$\vee$	or
$\neg$	not
$\rightarrow$	if . . . then
$\leftarrow$	. . . if . . .
$\leftrightarrow$	. . . iff . . .

Molecular propositions are obtained by combining atomic propositions by means of propositional connectives. For instance consider the following propositions:

[a: Freddie Mercury is a singer]

[b: Freddie Mercury is a guitarist]

[c: Freddie Mercury is a drummer]

[d: Freddie Mercury is a musician]

Combining them through connectives a logical formula can be obtained, such as:

$(a \vee b \vee c) \rightarrow d$

The distinguishing feature of the propositional connectives in classical logics is that they are truth-functional: this means that the truth or falsity of every composite proposition only depends upon the truth or falsity of the component propositions. Thus, every propositional connective is a truth function: it provides a single truth value (true or false) for every combinations of truth values of the propositions it connects. This means that in the end all composite propositions are truth-functions of elementary propositions. As [Wittgenstein, 1922] puts it, “when we know the truth values of the atomic propositions, we can determine the truth value of any combination of them”.

The function realized by each connective in propositional logic can be specified by means of a truth table, which indicates, for each combination of truth values of the connected propositions, the truth value of their composition.

Conjunction - In classical logic, conjunction is represented with the symbol $\wedge$. This connective corresponds to the English conjunction “and”. Thus the sentence:

Freddie Mercury is a singer and Brian May is a guitarist

can be reformulated as:

[a: Freddie Mercury is a singer] ∧ [b: Brian May is a guitarist]

using the letters this is:

a ∧ b

Conjunction is a dyadic operator: it applies to any two propositions A and B and joins them into compound proposition A ∧ B. The connective ∧ assigns to A ∧ B a truth value that depends upon (is a function of) the truth values of A and B: A ∧ B true if and only if, both A and B are true. Thus, its meaning is expressed by the truth table of Figure 1.3.

A	B	A ∧ B
1	1	1
1	0	0
0	1	0
0	0	0

Fig. 1.3 – A truth table. Each column of the table indicates truth values (0 or 1, denoting falsity or truth) for the formula on the top of it. The table shows how the truth value of the component propositions determines the truth value of the compound proposition. Each couple of truth values for the component propositions describes a possible state of affairs with regard to the concerned proposition, also called a case or a "truth possibility" for them (see [Wittgenstein, 1922]). The last column determines the truth possibility of the compound proposition, depending on what possible state of affair is realized. This truth table represents the truth possibilities for the conjunction operator.

Negation - Logical negation is represented by the symbol ¬, which can be read as "not" or "it is not the case that". Thus the natural language sentence:

Freddie Mercury is not a guitarist

can be reformulated as

¬ [a: Freddie Mercury is a guitarist]

In the abbreviated form

¬a

Please note that, in natural language, negation goes inside the sentence to which it applies, which is not the case in usual logical language. This corresponds to a general feature of logical syntax: It is compositional, in the sense that it allows building larger units by combining smaller units, without modifying the latter (the context where a certain syntactic structure appears does not modify its form). This contributes to making logical syntax much simpler than the syntax of natural language, more precise and easier to control. On the other hand, this leads to awkward and redundant expressions. More generally, every logical formalism tends to be more rigid than natural language, but it often enables to achieve more clarity and sometimes it facilitates the expression of ideas that it would be very difficult to articulate in natural language. The operator ¬ is monadic, namely, it applies to a single proposition: ¬A is true if and only if A is false. Thus its meaning is expressed through the truth table in Figure 1.4.

A	¬A
1	0
0	1

Fig. 1.4 – The truth table for negation

Disjunction - Disjunction is represented through the symbol ∨, which corresponds to the English conjunction "or". Thus the natural language proposition:

Freddie Mercury is a singer or a guitarist

can be reformulated as:

[a: Freddie Mercury is a singer] ∨ [b: Freddie Mercury is a guitarist]

in the abbreviated form:

a ∨ b

Like conjunctions, also disjunction is a dyadic operator and a truth function: A_B true if and only if at least one of A and B is true. Thus ∨ is characterized by the truth table in Figure 1.5.

A	B	A ∨ B
1	1	1
1	0	1
0	1	1
0	0	0

Fig. 1.5 – The truth table for disjunction

Thus the formula will be false only in case that Freddie Mercury is neither a singer nor a guitarist. It will be true in all other possible cases. The concept of disjunction just described is the so called *weak disjunction*, which the Romans expressed with the word *vel* (this is why the symbol ∨, similar to a "v", is used to express this idea). However in English the word *or* is also (though more rarely) used, often preceded by *either*, to express a different, more specific (stronger) notion of disjunction: the idea that only one of the disjoined proposition is true, not both of them. Consider the following proposition:

Either Freddie Mercury is a singer or he is a guitarist

This seems to assert that he has one musical talent but no more. Similarly consider the prescription

It is allowed to take either the fruit or the dessert

This permission does not seem to cover the case where one takes both the fruit and the dessert (the addressee of the prescription should consider that this remains forbidden). It is often said that the Romans used the term "*aut*" to express the second kind of disjunction. In logic the strong disjunction is also called xor, and is expressed with the symbol ⊻. Thus

[a: Freddie Mercury is a singer] ⊻ [b: Freddie Mercury is a guitarist]

is true if Freddie Mercury is just one of those, it is false if he is both. The functioning of ⊻ is expressed by the truth table in Figure 1.6.

A	B	A ⊻ B
1	1	0
1	0	1
0	1	1
0	0	0

Fig. 1.6 – The truth table for the strong disjunction

Material Conditional - The symbol → represents material conditional, which corresponds to the English conjunction "if...then…". Thus the conditional

If you don't like fruit, then take the dessert

can be reformulated as

[a: you don't like fruit] → [b: you take the dessert]

Using the abbreviation:

a → b

Similarly, the conditional

if a plane hit the Pentagon then plane debris can be found around the impact area

can be reformulated as

[a: a plane hit the Pentagon] → [b: plane debris can be found around the impact area]

Like conjunction and disjunction, material conditional is a truth functional dyadic operator: A → B is false if and only if A is true and B is false. Thus → is characterized by the truth table in Figure 1.7.

A	B	A → B
1	1	1
1	0	0
0	1	1
0	0	1

Fig. 1.7 – The truth table for the material conditional.

Thus in our example, a plane hitting the Pentagon is sufficient condition for plane debris to be found around the impact area, or equivalently, that plane debris being found around the impact area is a necessary condition of the plane having hit the Pentagon. There is a mismatch between the logical equivalence and our linguistic intuitions, since speaking of a condition usually means that an event or situation which precedes the conditioned event, but this is not captured by propositional logic.

Biconditional - The symbol ↔ is used to represent material biconditional, to be read as "if and only if". Thus the conditional

Socrates will drink the poison if and only if he is condemned by the Athenians.

can be reformulated as

[Socrates will drink the poison] ↔ [Socrates is condemned by the Athenians]

Also the biconditional is a truth functional dyadic operator: A ↔ B is true if and only if both A and B are true or both of them are false. Thus ↔ is characterized by the truth table in Figure 1.8.

A	B	A↔B
1	1	1
1	0	0
0	1	0
0	0	1

Fig. 1.8 – The truth table for the biconditional.

1.2.2.2. Considerations on Propositional Logic and Legal Reasoning

It is now the time to take a first step from logics into the legal field. Comparing this logical feature with legal mechanics, we realize that a material conditional does not fully capture the connection between a legal antecedent and a legal conclusion, this being prevented by two features of legal norms: legal norms are both counterfactual and defeasible. This does not mean, however, that we cannot use classical logic for modelling legal norms: we must be aware of the limits of the formal tool we are using.

An interesting feature of classical propositional logic is the possibility to infer whatever conclusion from a contradiction (from a falsity anything follows, in Latin *ex falso sequitur quodlibet*). This means that in propositional logic contradictions do not remain isolated, but rather sprawl arbitrary conclusions. As noted in [Sartor 2005], in the legal field it is instead necessary to keep contradictions local, and to solve them without getting to further absurdities. Assume for instance to have two conflicting rules, one saying that citizens have the right to vote, and the other saying that people of minor age do not have the right to vote. Translating them into propositional logic:

[a: the person is a citizen] → [b: the person has the right to vote]

[c: the person is of minor age] → ¬[b: the person has the right to vote]

Assume also that, as a matter of fact, the person considered is both a citizen and of minor age. The following derivation is obtained:

a (premise)

a → b (premise)

c (premise)

c → ¬b (premise)

b (from 1 and 2 by →)

¬b (from 3 and 4 by →)

? (from 3 and 6 by →)

[d: every citizen is a senator] (from 7 by ?)

Thus an arbitrary formula *d* can be derived from the inconsistency between *b* and ¬*b*. Clearly, this is not the right way to address inconsistencies in the law. Nobody would accept this argument, though it appears to be semantically sound (in all cases in which the inconsistent premises are true, that is never, also the consequence *d* is true).

1.2.2.3. Predicate Logic

Predicate logic is built on top of propositional logic, to which it adds the feature of formal structure analysis for atomic propositions. To use the metaphor of [Frege, 1970], “atomic propositions can be looked into only when we take the microscope of predicate logic". In particular, using predicate logic, the following elements of a sentence are identified:

- **Predicates**, which express actions, properties and relationships;
- **Terms**, which indicate the objects to which predicates refer.

For instance, in the propositional formula:

Socrates drinks Poison

predicate logic allows to further define the proposition by distinguishing the predicate

1 drinks 2

having two arguments, indicated by the numbered place-holders, and terms Socrates and poison. By substituting the terms for the place-holders in the predicate, as suggested by [Quine, 1974], the result is propositional formula. In the usual, compact logical syntax, the predicate precedes its arguments:

drinks(Socrates;Poison)

The distinction between terms and predicates is arbitrary rather than fixed, and depends on the opportunity in the single case: if we were interested only in establishing who drinks Poison, then we could have used the following predicate

1 drinks Poison

In propositional logic there are two kinds of terms: constants and variables. Constants refer to specific individuals, while variables refer to any object within the domain of discourse. Variables are indicated with letters x, y, w, z. For instance,

[x drinks y]

or

Drinks(x; y)

is a formula expressing that an undetermined individual, referred to as x, sings an undetermined object, referred to as y.

A formula like this does not yet express a proposition: as seen in propositional logic, a proposition is an entity which is subject to be true or false, but it cannot be established whether the present formula is true or false until the variables are specified. To transform a formula containing variables into a propositional formula individual names for the variables have to be substituted, as in the following example:

Socrates drinks Poison

Predicate logic offers also another way to refer to individual entities, namely by using function symbols. A function identifies an individual entity on the basis of its connection with other entities. For instance, the function mother(x) denotes a particular person (a mother) on the basis of the fact that it has a particular connection (motherhood) with another individual (a child). For instance, mother(Sarah) identifies a particular woman (Sarah's mother), on the basis of the fact that she is is the unique mother of Sarah.

1.2.2.4. Quantifiers

There is a second way to obtain a propositional formula out of an expression containing variables. This consists in using two typical constructs of predicate logic, the universal quantifier FORALL, also denoted through the symbol ∀ and the existential quantifier FORSOME (to be read as "it exists an"), also denoted through the symbol ∃. Let us consider first the universal quantifier. The very general proposition expressing that any object is a thing can be expressed as follows:

FORALL (x) [x is a thing]

which in the symbolism of mathematical logic becomes:

$\forall x(Thing(x))$

Very little can be said, in general, of all the possible objects. More interesting assertions can be obtained by using universal quantification over conditionals. In this case it can be specified that all objects which satisfy a certain condition have certain properties:

FORALL (x)

IF [x is a citizen] THEN [x has the right to vote]

which in the symbolism of mathematical logic becomes:

$\forall x(Citizen\ (x) \rightarrow RightToVote(x))$

meaning that everyone who is a citizen has the right to vote. The existential quantifier FORSOME is used to express the idea that there exists at least one individual that satisfies a certain predicate. For example:

FORSOME (x) [x; is Senator of France]

expresses the idea that there is at least one entity, which is Senator of France (France has at least one Senator). In the usual syntax of predicate logic it becomes the proposition:

$\exists x(SenatorOf(x;France)$

similarly, the following proposition

FORALL (y) IF[y is a person]

THEN FORSOME (x) [x is y's mother]

in the logical symbolism

$\forall y(Person(y) \rightarrow \exists x Mother(x;\ y))$

shows how one can combine the two quantifiers. It is to be read as: “for any entity, if that entity is a person then some other entity is the mother of that person” or more simply “every person has a mother”.

1.2.2.5. Horn Clause Logic

The logician Alfred Horn, investigating the mathematical properties of sentences with at most one positive literal in [Horn 1951], highlighted some features which are now largely exploited in logic programming and constructive logic.

A Horn clause is a clause with at most one positive literal. Any Horn clause therefore belongs to one of four categories:

- A **rule**: 1 positive literal, at least 1 negative literal. A rule has the form:

  $\neg P1 \vee \neg P2 \vee ... \vee \neg Pk \vee Q$

  This is logically equivalent to

  $(P1 \wedge P2 \wedge ... \wedge Pk) \rightarrow Q$

  thus, an *if…then* implication with any number of conditions but one conclusion.

  Examples:

  - ¬man(x) ∨ mortal(x) (All men are mortal);
  - ¬parent(x,y) ∨ ¬ancestor(y,z) ∨ ancestor(x,z) (If x is a parent of y and y is an ancestor of z then x is an ancestor of z.).

- A **fact** or unit: 1 positive literal, 0 negative literals.

  Examples:

  - man(socrates);
  - parent(elizabeth,charles);
  - knows(x,x) (Everyone knows himself – in a trivial sense).

- A **negated goal**: 0 positive literals, at least 1 negative literal. In virtually all implementations of Horn clause logic, the negated goal is the negation of the statement to be proved; the knowledge base consists entirely of facts and goals. The statement to be proven, therefore, called the goal, is a single unit or the conjunction of units; an existentially quantified variable in the goal turns into a free variable in the negated goal. E.g. If the goal to be proven is

  ∃x(male(x) ∧ ancestor(elizabeth, x)) (show that there exists a male descendent of Elizabeth)

  the negated goal will be

¬male(x) ∨ ¬ancestor(elizabeth,x).

- The **null clause**: 0 positive and 0 negative literals. Appears only as the end of a resolution proof.

Horn clause logic is equivalent in computational power to a universal Turing machine. As long as resolution is restricted to Horn clauses, some interesting properties apply:

- When resolving Horn clauses A and B to get clause C, then the positive literal of A will resolve against a negative literal in B, so the only positive literal left in C is the one from B. Thus, the result of two Horn clauses is a Horn clause;

- When resolving a negated goal G against a fact or rule A to get clause C, the positive literal in A resolves against a negative literal in G. C is left with no positive literal, and thus is either a negated goal or the null clause;

- When trying to prove G from D, where ¬G is a negated goal, and D is a knowledge base of facts and rules, and using the set of support strategy in which no resolution ever involves resolving two clauses from D together, then every resolution combines a negated goal with a fact or rule from D and generates a new negated goal. Taking a resolution proof, and tracing the way back from the null clause at the end to ¬G at the beginning (since every resolution involves combining one negated goal with one clause from D) it is clear that the sequence of negated goals involved can be linearly ordered. That is, the final proof, ignoring dead ends has the form:

  ¬G resolves with C1 from D, generating negated goal P2

  P2 resolves with C2 from D, generating negated goal P3

  ...

  Pk resolves with C2 from Gamma, generating the null clause.

Therefore, the process of generating the null clause can be viewed as a state space search where:

- A state is a negated goal;

- An operator on negated goal P is to resolve it with a clause C from D;

- The start state is ¬G;

- The goal state is the null clause.

Propositional Horn clauses are also of interest in computational complexity, where the problem of finding truth value assignments to make a conjunction of propositional Horn clauses true is a P-complete problem (solvable in linear time), sometimes called HORNSAT as in [Cook and Nguyen 2010].

[Van Emden and Kowalski 1976] investigated the model theoretic properties of Horn clauses in the context of logic programming, showing that every set of definite clauses has a unique minimal model. An atomic formula is logically implied by the definite clauses if and only if it is true in this model. It follows that a goal clause is logically implied by a set of definite clauses if and only if it is true in the minimal model. The minimal model semantics of Horn clauses is the basis for the stable model semantics of logic programs.

1.2.3. Modal Logic

Modal logic is the study of modal propositions and of their logical relationships. The basic modal propositions concern necessity and possibility. For example, the following are modal propositions:

- It is possible that it will be sunny tomorrow;
- It is possible for humans to travel to the center of the Earth;
- It is not possible that: every person is mortal, Socrates is a person, and Socrates is not mortal;
- It is necessary that either it is raining here now or it is not raining here now;
- A proposition p is forbidden if and only if the negation of p is obligatory.

Modal logic was first discussed in a systematic way by Aristotle in *De Interpretatione*. Aristotle noticed that the notions of necessity and possibility were interdefinable. The proposition *p* is possible may be defined as: *not-p* is not necessary. Similarly, the proposition *p* is necessary may be defined as: *not-p* is not possible. Aristotle also pointed out that from the separate facts that *p* is possible and that *q* is possible, it does not follow that the conjunctive proposition *p* and *q* is possible.

As [Lemmon and Scott 1977] reports, philosophers after Aristotle added other interesting observations to the long list of implications spawning from these considerations. Contributions were made, among others, by the Megarians, the Stoics, Ockham, and Pseudo-Scotus. Besides an incursion on the field by Leibniz (described in [Poser 1969]), interest in modal logic resumed only in the twentieth century with Clarence Irving Lewis' search for an axiom system to characterize `strict implication' in [Lewis 1918]. As explained in the next paragraph, G. Henrik von Wright was among the principal developers of deontic systems of modal logic. The model-theoretic study of the logical consequence relation in modal logic began with [Carnap 1947]. Further innovations in modal logic (among which the introduction of the domain of possible worlds) were developed in [Kripke 1959], though they were anticipated in [Kanger 1957] and [Hintikka 1957].

The operators "*it is possible that*" and "*it is necessary that*" are called "*modal*" operators, because they specify a way (*modus*) in which the rest of the proposition can be said to be true. There are other modal operators, for example: it once was the case that, it will once be the case that, and it ought to be the case that. Moreover, certain modal propositions logically imply others. For example, the proposition *it is necessary that p* logically implies the proposition that *it is possible that p*, but not vice versa. These implications enhance the intuitive understanding of the modal propositions involved, because to understand a proposition is, in part, to grasp what it logically implies.

Models provide interpretations for the language. A model M for the language is typically defined to be a triple <*W, R, V*> where W is a nonempty set of possible worlds, R the accessibility relation, and V a valuation function that assigns to each atomic sentence *p* a set of worlds V(*p*). These models allow to define the model-theoretic notions of truth, logical truth, and logical consequence. Whereas truth and logical truth are semantic properties of the sentences of the language, logical consequence is a semantic relation among sentences. A sentence is said to be logically true, or valid, only if it is true in all models, and it is said to be valid with respect to a class *C* of models only if it is valid in every model in the class. The proof theory proceeds along similar lines: rules of inference relate certain sentences to others, indicating which sentences can be inferred, and from which sentence. A logic Σ is a set of sentences delimited by the rules of inference defining that logic. A theorem of a logic is simply a sentence inside Σ. A logic Σ is said to be sound with respect to a class of models C just in case every sentence that is a theorem of Σ is valid with respect to the class C, and a logic Σ is said to be complete with respect to a class C of models just in case every sentence that is valid with respect to C is a theorem of Σ.

1.2.4. Deontic Logic

In the last 60 years there has been a large amount of research on formal models of normative concepts. This is an interdisciplinary domain, where philosophers like [Conte et al., 1977] logicians like [von Wright, 1951]), legal theorists like [Alchourron, 1996], and computer scientists like [McCarthy, 1986] have merged their efforts.

Any logical analysis of the normative legal concepts requires a formal modelling of deontic notions, such as obligation (duty) and permission. These ideas have been applied by using different logical tools, most frequently related to the possible-worlds semantics of modal logic.

1.2.4.1. Obligatory

We can represent obligations through formulas having the following structure:

Obl A

which stands for "it is obligatory that A", where A denotes any action or state of affairs, and Obl is the deontic operator for obligation, to be read as "it is obligatory that." For instance, the following formula states that Sarah has the obligation to pay taxes:

Obl [Sarah pay taxes]

1.2.4.2. Forbidden

As intuitively expected, when one is obliged not to perform a certain action he is forbidden from doing that action. Here is an example:

Forb [Sarah download copyrighted work]

which stands for "it is forbidden that Sarah downloads copyrighted work".

1.2.4.3. Permission

The operator Perm is used to express permissions. For example, to indicate that Sarah is permitted to access the Springer Online Catalogue:

Perm [Sarah access Springer Online Catalogue]

which stands for "it is permitted that Sarah accesses the Springer Online Catalogue".

1.2.4.4. Hohfeldian relations between deontic operators

Following Hohfeld (see 1.1.8.), in deontic logic the concept of permission is defined as the opposite of the one of obligation. It seems intuitively obvious that:

Forb A ≡ Obl ¬A

which stands for "Being prohibited to perform an action means being obliged not to do it".

For instance, that Sarah is forbidden from smoking means that she is obliged not to smoke. Likewise, that some A is permitted means that A is not forbidden:

Perm A ≡ ¬Forb A

then:

Perm A ≡ ¬Obl ¬A

The deontic qualifications "obligatory" and "forbidden" are complete, in the sense that they determine the deontic status of both the action or state of affairs A they are concerned with and the complement of A: to say that the action or state of affairs A is obligatory is equivalent to saying that ¬A is forbidden, and to say that A is forbidden is equivalent to say that ¬A is obligatory.

On the contrary, to say that an action or state of affairs is permitted does not add information about the status of its complement. In particular, when a positive action or state of affairs is permitted (namely, the action is not forbidden), then its negation can be likewise permitted or it can be forbidden (this will be the case when the action, besides being permitted, also is obligatory).Consider for example, "*wearing a full helmet*", abbreviated as H:

Situation	wearing a helmet (H)	not wearing a helmet (¬H)
Inside the Parliament	Forb H	Obl ¬H
Riding a motorbike	Obl H	Forb ¬H
Riding a bike	Perm H	Perm ¬H

Fig. 1.9 – Complete deontic qualifications

The omission of the helmet (¬H) is permitted when riding a bike, but it is forbidden when riding a motorbike, and is obligatory inside the Parliament. From the table 1.x it appears that to express the normative qualification of wearing a helmet while riding a motorbike or inside the parliament, it is sufficient to say that while riding a motorbike wearing the helmet is obligatory while inside the Parliament it is forbidden.

On the contrary, saying that H is permitted (namely, not prohibited) when riding a bike is not sufficient to fully specify H's normative status in that case: the permission to wear a helmet (Perm H) is consistent both with the permission not to wear it (Perm ¬H) and with the prohibition not to wear it (with Forb ¬H), that is, with the obligation to wear it (with Obl H). Thus, to provide a complete deontic specification, it is necessary to specify whether not wearing the helmet is forbidden or permitted. When riding a bike, wearing a helmet is permitted (as when riding a motorbike, and contrary to what is the case when inside a Parliament), but not wearing the helmet is permitted too (as when inside a Parliament, and contrary to what is the case when riding a motorbike). In conclusion, besides an action or state of affairs being obligatory (and its negation being forbidden), and besides its being forbidden (and the negation being obligatory), there is a third way for the deontic status of an action or state of affairs to be fully specified: this consists in both the action or state of affairs being permitted and its negation being permitted. In common speech, when one says "permitted," one usually refers to this third option.

1.2.4.5. Facultative

Any A is facultative when both A and ¬A are permitted:

$$\text{Facult } A \equiv (\text{Perm } A \wedge \text{Perm } \neg A)$$

which stands for "it is facultative that A is equivalent to it is permitted that A and it is permitted that ¬A.

For example, saying that when riding a bike it is facultative to wear the helmet corresponds to saying that it is permitted both to wear the helmet and to not wear it. Note that something being facultative does not entail that others are forbidden to prevent it (or that other are forbidden to prevent its negation. In this sense, faculty is a weak notion of freedom.

In deontic logic the permission that one agent *j* does some A has to be distinguished from the prohibition that another (or all others) prevents *j* from doing this A: it is possible that one is permitted to do actions that others are permitted to prevent. Otherwise, there are general prohibitions upon others that—by limiting in general their action—prescribe certain ways of interfering with the holder of a mere permission, and thus indirectly provide a certain legal protection for the possibility of doing the permitted action. As in [Hart 1982], at least the cruder forms of interference, such as those involving physical assault or trespass, will be criminal or civil offences or both, and the duties or obligations not to engage in such modes of interference constitute a protective perimeter behind which liberties exist and may be exercised.

[Alexy, 1985] refers to such faculties by speaking of directly protected freedoms, as opposed to unprotected freedoms, which are not accompanied by the prohibition on interference. Among such directly protected freedoms are the negative liberties one has towards the State in liberal countries (for instance, freedom of speech, of religion, and so on). Thus a protected freedom, with regard to action A would be expressed as:

Facult A = Forb [prevent A] $\wedge$ Forb [prevent $\neg$A]

An even stronger notion of one's liberty to do A is obtained when the others' prohibition from interfering with A is coupled with the obligation (upon others or upon the government) to provide some means for performing A, namely, the obligation to ensure that the concerned agent has the effective capability of doing what he or she is permitted to do (in this way, the so-called negative freedom becomes as well a positive or substantive freedom; see [Sen, 1999]). But in order to capture all aspects of facultativeness it would be necessary to represent and reason about actions and to introduce a logic of agency, which is outside the scope of the present research.

1.2.4.6. Bentham's Logic of will

The formalization presented in the last section relies on formulations by Jeremy Bentham made way before the authors above formalized deontic logics. Bentham came to these considerations by investigating the essential elements of law. According to Bentham, "There are two things essential to every law: an act of some sort or other, being the object of a wish or volition on the part of the legislator; and a wish or volition of which such act is the object."

In Bentham's thought legal science needed a particular logic, different from the Aristotelian logic of the understanding. He calls this new logic the *logic of imperation* or *logic of the will*. In chapter ten of [Hart 1970] Bentham presents these aspects, more than one hundred fifty years before Von Wright:

> "Concerning an act, the aspect is either decided or directive; either neutral or undecided. If it is decided, this is a command or a prohibition which expresses the wish of the legislator to see people to whom the legal norm is addressed perform or refrain to perform a specific act. If it is neutral, it is a non-command or a nonprohibition, which is the equivalent of permission: the legislator does not command anything to his subjects concerning a specific conduct to adopt." [Hart 1970]

These different aspects of the will act like a deontic operator in contemporary logic and this is exactly what Bentham has in mind. As for von Wright, symbols following a deontic operator denote for Bentham properties for class of acts like stealing, killing, smoking etc. Bentham's example can be represented like this:

Every man shall wear a helmet = Obl H.

No man shall wear a helmet = Forb H.

Any man may avoid to wear a helmet = $\neg$ Obl H.

Any man may wear a helmet = Perm H.

Bentham suggests a parallelism between relations that can be established between these statements and the relations of conversion existing between classical logical statements as they appear in a logical square: as in classical logics two contradictory statements cannot be simultaneously true or false, also two contradictory directive statements cannot be simultaneously valid or invalid. Therefore, Obl H and ¬ Obl H can't be valid simultaneously but one of them must be valid. That is:

¬ (Obl H & Obl H) & (Obl H v ¬ Obl H)

similarly concerning Forb H and ¬ Forb H:

¬ (Forb H & Forb H) & (Forb H v ¬ Forb H)

If Bentham's proposal is a system of axiomatic calculus of deontic logic obtained through two definitions, one axiom, and rules of propositional calculus. The two definitions in Bentham’s sentence “a negative aspect towards a positive act is equipollent to an affirmative aspect towards the correspondent negative act”. These two definitions can be obtained from the sentence:

Forb H if and only if Obl ¬H

¬Obl H if and only if Perm ¬H

The sole axiom is the following:

Obl H implies Perm H

Adding to this basis the following passage of [Hart 1970], which can be formalized to obtain other theorems that can be demonstrated, it is possible to foresee the possibility of formalization offered by Bentham:

> “First, it may be commanded: it is then left unprohibited: and it is not prohibited nor left uncommanded. 2. It may be prohibited: it is then left uncommanded: and it is not commanded nor permitted (that is left unprohibited). 8. It may be left uncommanded: it is then not commanded: but it may be either prohibited or permitted: yet so as that if it be in the one case it is not in the other. 4. It may be permitted: it is then not prohibited: but it may be either commanded or left uncommanded: yet so as that if it be in the one case, it is not in the other, as before.” [Hart 1970]

For example: ‘it may be commanded: it is then left unprohibited’ can be formalized as:

OblH → ¬ForbH.

It is then clear that Bentham's logic of the will is a deontic logic, and that, when formalized, it leads to a deontic calculus. It is a logic of affirmative statements concerning prescriptive statements. Legal validity means belonging to the field of law. One should understand that Bentham's interest goes beyond an attempt of formalization, even in common language and even it creates a great theoretical field. The logic of the will finds a place in a broader project of reforming English law at the level of his vocabulary and his grammar, which is within the scope of Benthamian jurisprudence. For more on the subject see [Bozzo-Rey 2009].

1.2.5. Description Logics

The term Description Logics (DLs) doesn't define a logics formalism but rather a family of knowledge representation (KR) formalisms that represent the knowledge of an application domain by first defining the relevant concepts of the domain, and then using these concepts to specify properties of objects and individuals occurring in the domain. Description Logics are descended from so-called "structured inheritance networks" [Brachman 1977, 1978], which were introduced to overcome the ambiguities of early semantic networks and frames, and which were first realized in the system Kl-One [Brachman and Schmolze 1985]. The following ideas, first put forward in Brachman's work on structured inheritance networks, have largely shaped the subsequent development of DLs:

- The basic syntactic building blocks are atomic concepts (unary predicates), atomic roles (binary predicates), and individuals (constants);
- The expressive power of the language is restricted in that it uses a rather small set of (epistemologically adequate) constructors for building complex concepts and roles.

The characteristics of the Description Logics language include formal, logic-based semantics and the emphasis on reasoning as a central service: reasoning allows one to infer implicitly represented knowledge from the knowledge that is explicitly contained in the knowledge base. The two main inference patterns supported by Description Logics are classification of concepts and individuals, described in [Baader and Nutt 2003].

- Classification of **concepts** determines subconcept/superconcept relationships (called subsumption relationships in DL) between the concepts of a given terminology, and thus allows one to structure the terminology in the form of a subsumption hierarchy. This hierarchy provides useful information on the connection between different concepts, and it can be used to speed-up other inference services;
- Classification of **individuals** (or objects) determines whether a given individual is always an instance of a certain concept (i.e., whether this instance relationship is implied by the description of the individual and the definition of the concept). It thus provides useful information on the properties of an individual. Moreover, instance relationships may trigger the application of rules that insert additional facts into the knowledge base.

Implicit knowledge about concepts and individuals can be inferred automatically with the help of inference procedures. In particular, subsumption relationships between concepts and instance relationships between individuals and concepts play an important role: unlike IS-A links in Semantic Networks, which are explicitly introduced by the user, subsumption relationships and instance relationships are inferred from the definition of the concepts and the properties of the individuals.

1.2.5.1. The issue of computability

Because Description Logics are a KR (Knowledge Representation) formalism, and being assumed that a KR system should always answer the queries of a user in reasonable time, the reasoning procedures DL researchers are interested in are decision procedures, which means they should always terminate, both for positive and for negative answers. Since the guarantee of an answer in finite time does not imply that the answer is given in reasonable time, investigating the computational complexity of a given DL with decidable inference problems is an important issue. Decidability and complexity of the inference problems depend on the expressive power of the DL at hand. On the one hand, very expressive DLs are likely to have inference problems of high complexity, or they may even be undecidable. On the other hand, very weak DLs (with efficient reasoning procedures) may not be sufficiently expressive to represent the important concepts of a given application. Investigating this trade-off between the expressivity of DLs and the complexity of their reasoning problems has been one of the most important issues in DL research.

The first formal investigations of the computational properties of the languages used in early DL showed that these languages were too expressive, which led to undecidability of the subsumption problem [Schmidt-Schauß 1989; Patel-Schneider 1989]. The first worst-case complexity results [Levesque and Brachman 1987; Nebel 1988] showed that the subsumption problem is intractable (i.e., not polynomially solvable) even for very inexpressive languages. Intractability of reasoning, however, does not prevent a DL from being useful in practice, provided that sophisticated optimization techniques are used when implementing a system based on such a DL, described in [Horrocks 2003]. When implementing a DL system, the efficient implementation of the basic reasoning algorithms is not the only issue, though. On the one hand, the derived system services (such as classification, i.e., constructing the subsumption hierarchy between all concepts defined in a terminology) must be optimized as well [Baader et al. 1994]. On the other hand, one needs a good user and application programming interface. Most implemented DL systems provide for a rule language, which can be seen as a very simple, but effective, application programming mechanism.

1.2.5.2. Building Knowledge Bases

A KR system based on Description Logics provides facilities to set up knowledge bases, to reason about their content, and to manipulate them. A knowledge base (KB) comprises two components, the TBox and the ABox. The TBox introduces the terminology (the vocabulary) of an application domain, while the ABox contains assertions about named individuals in terms of this vocabulary.

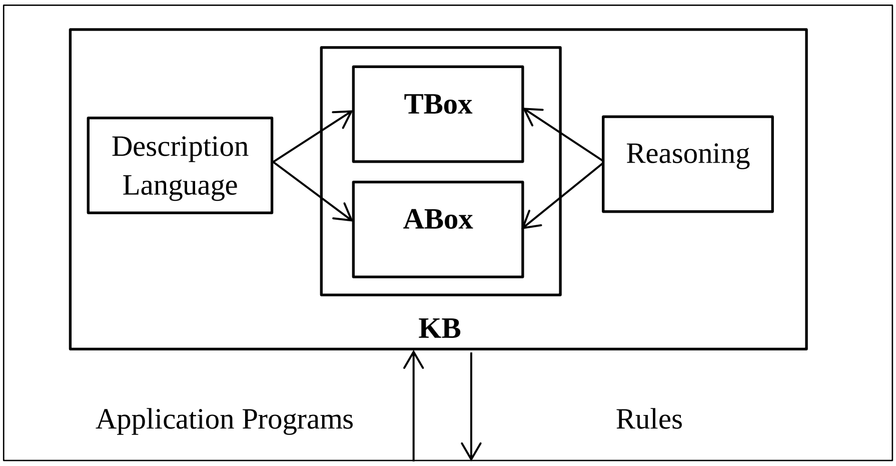


Figure 1.10: Architecture of a knowledge representation system based on Description Logics.

The vocabulary consists of concepts, which denote sets of individuals, and roles, which denote binary relationships between individuals. In addition to atomic concepts and roles (concept and role names), all DL systems allow their users to build complex descriptions of concepts and roles. The TBox can be used to assign names to complex descriptions. The language for building descriptions is a characteristic of each DL system, and different systems are distinguished by their description languages. The description language has a model-theoretic semantics. Thus, statements in the TBox and in the ABox can be identified with formulae in first-order logic or, in some cases, a slight extension of it. A DL system not only stores terminologies and assertions, but also offers services that reason about them. Typical reasoning tasks for a terminology are to determine whether a description is satisfiable (i.e., non-contradictory), or whether one description is more general than another one, that is, whether the first subsumes the second. Satisfiability checks of descriptions and consistency checks of sets of assertions are useful to determine whether a knowledge base is meaningful at all. With subsumption tests, one can organize the concepts of a terminology into a hierarchy according to their generality.

1.2.5.3. Description languages

Elementary descriptions are atomic concepts and atomic roles. Complex descriptions can be built from them inductively with concept constructors, and a set of constructors defines a description language. The language $\mathcal{AL}$ (= attributive language) has been introduced in [Schmidt-Schauß and Smolka 1991] as a minimal language that is of practical interest. The other languages are extensions of it. To give examples of what can be expressed in $\mathcal{AL}$, let's suppose that Person and Female are atomic concepts. Then Person $\wedge$ Female and Person $\wedge$ $\neg$Female are $\mathcal{AL}$ concepts describing, intuitively, those persons that are female, and those that are not female. In addition, the supposition that hasChild is an atomic role allows forming the concepts:

Person $\wedge$ $\exists$ hasChild and

Person $\wedge$ $\forall$hasChild.Female,

denoting those persons that have a child, and those persons all of whose children are female. Using the bottom concept ($\perp$), the persons without a child can be described by the concept:

Person $\wedge$ $\forall$hasChild.$\perp$

The next step is to introduce interpretations $\mathcal{I}$ consisting in a non-empty set $\Delta(\mathcal{I})$ and a function assigning to each atomic concept A a set $A(\mathcal{I}) \subseteq \Delta(\mathcal{I})$ (interpreted atomic concepts as subset of the interpretations domain), and to each atomic role R a binary relation $R(\mathcal{I}) \subseteq \Delta(\mathcal{I}) \times \Delta(\mathcal{I})$. The interpretation function is extended with the following basic definitions:

$\top(\mathcal{I})$	=	$\Delta(\mathcal{I})$
$\perp(\mathcal{I})$	=	0
$(\neg A)(\mathcal{I})$	=	$\Delta(\mathcal{I}) \setminus A(\mathcal{I})$
$(C \wedge D)(\mathcal{I})$	=	$C(\mathcal{I}) \wedge D(\mathcal{I})$

$(\forall R.C)(\mathcal{I}) \quad = \quad \{a \in \Delta(\mathcal{I}) \mid \forall b.\ (a, b) \in R(\mathcal{I}) \rightarrow b \in C(\mathcal{I})\}$

$(\exists R.T)(\mathcal{I}) \quad = \quad \{a \in \Delta(\mathcal{I}) \mid \exists b.\ (a, b) \in R(\mathcal{I})\}$

We say that two concepts (C and D) are equivalent if C($\mathcal{I}$) D($\mathcal{I}$) for all interpretations $\mathcal{I}$. For example, one can easily verify that

∀hasChild.Female ∧ ∀hasChild.Student

and

∀hasChild.(Female ∧ Student)

are equivalent.

More expressive languages are obtained adding further constructors to $\mathcal{AL}$.

- The **union of concepts**, indicated by the letter $\mathcal{U}$;
- **Full existential quantification**, indicated by the letter $\mathcal{E}$;
- **Number restrictions**, also called cardinality restrictions and indicated by the letter $\mathcal{N}$;
- The **complex concept negation**, indicated by the letter $\mathcal{C}$, for "complement".

Extending $\mathcal{AL}$ by any subset of the above constructors yields a particular $\mathcal{AL}$ language. Each $\mathcal{AL}$ language is named with a string of the form

$\mathcal{AL}[\mathcal{U}][\mathcal{E}][\mathcal{N}][\mathcal{C}]$;

where a letter in the name stands for the presence of the corresponding constructor. Other relevant extensions are:

- **Functional Properties** indicated by the letter $\mathcal{F}$;
- **Role Hierarchy** indicated by the letter $\mathcal{H}$;
- **Limited Complex Role Inclusion Axioms** (reflexivity and irreflexivity; role disjointness), indicated by the letter $\mathcal{R}$;
- **Nominals** (enumerated classes of object value restrictions), indicated by the letter $\mathcal{O}$;
- **Inverse Properties**, indicated by the letter $\mathcal{I}$;
- **Qualified cardinality restrictions**, indicated by the letter $\mathcal{Q}$;
- **Use of datatype properties, data values or data types**, indicated by the letter ($\mathcal{D}$).

$\mathcal{ALC}$ is a centrally important description logic from which comparisons with other varieties can be made. $\mathcal{ALC}$ is simply $\mathcal{AL}$ with complement of any concept allowed (not just atomic concepts). $\mathcal{S}$ is an abbreviation for $\mathcal{ALC}$ with transitive roles. The Protégé ontology editor supports $\mathcal{SHOIN(D)}$. OWL 2 provides the expressiveness of $\mathcal{SROIQ(D)}$, OWL-DL is based on $\mathcal{SHOIN(D)}$, and for OWL-Lite it is $\mathcal{SHIF(D)}$.

1.2.6. Default Logic

Knowledge about a world is "almost" always true, as there is always some exceptions that could apply. For example, most people are liable to criminal laws except for heads of State, ambassadors and so on. Given a particular person, we will conclude that it is liable unless we happen to know that it satisfies one of these exceptions. How is the fact that most people are liable to be represented? The natural first order representation explicitly lists the exceptions to liability:

$$\text{Person}(x) \wedge \neg\text{HeadofState}(x) \wedge \neg\text{Ambassador}(x) \wedge \ldots = \text{Liable}(x)$$

The problem is that with this representation one cannot conclude of a "general" person that it is liable, if there is no further information besides the fact that it is a person we are talking about. Concluding that a general person is liable is prevented even though intuitively desirable. Something is required to allow a person to be liable by default.

Default Logic achieves this by introducing a statement such as: "If x is a person, then in the absence of any information to the contrary, infer that x is liable". The problem then is to interpret the phrase "in the absence of any information to the contrary". The interpretation adopted in Default Logic is "It is consistent to assume that x is liable". Thus "If x is a person and it is consistent to assume that x is liable, then infer that x is liable". This is represented more formally as the following default formula:

$$\frac{\text{Person}(x) : \mathbf{M}\text{Liable}(x)}{\text{Liable}(x)}$$

Here **M** is to be read as *it is consistent to assume*. The exceptions to liability are then given a standard first order representation.

$$\text{HeadofState}\ (x) = \neg\text{Liable}(x)$$

$$\text{Ambassador}(x) = \neg\text{Liable}(x)$$

Notice that if Liable(x) is inferred by default then the assertion Liable(x) has the status of a belief; it is subject to change, say by the subsequent discovery. The default rule can then reinterpreted as

> If x is a person and it is consistent to believe that x is liable then one may believe that x is liable.

Default logics was introduced in [Reiter 1980], which also provides an appropriate formal definition of the consistency requirement. Interestingly, in order to achieve its goals of default knowledge representation, the author defines the concept of closed world assumption, where failure to find a proof sanctions an inference. The concepts of open and closed world assumption will be presented in section 2.2.5.1.

1.2.7. Defeasible Logic

Any reasoning system that preserves truth must be monotonic, but a reasoning system that preserves justification will not be monotonic. That means that a belief that φ might be justified based on our belief in some set of propositions S, but there could be another set of propositions T such that if we came to believe all the propositions in S ∪ T we would no longer be justified in believing φ.

Defeasible logics were introduced by [Nute 1994]. A defeasible theory consists of five different kinds of knowledge: facts, strict rules, defeasible rules, defeaters, and a superiority relation.

Facts are indisputable statements, for example, “Socrates is a citizen”. Written formally, this would be expressed as

citizen(Socrates)

Strict rules are rules in the classical sense: whenever the premises are indisputable (e.g., facts) then so is the conclusion. An example of a strict rule is

Citizens are men

written formally

citizen(x) → man (x).

Defeasible rules are rules that can be defeated by contrary evidence. An example of such a rule is “Men typically speak”; written formally:

man (x) ⇒ speaks (x).

The idea is that if we know that something is a man, then we may conclude that it speaks, unless there is other, not inferior, evidence suggesting that it may not speak.

Defeaters are rules that cannot be used to draw any conclusions. Their only use is to prevent some conclusions. In other words, they are used to defeat some defeasible rules by producing evidence to the contrary. An example is “If a man is a savage then it might not be able to speak”. Formally:

savage (x) ⇝ speaks (x).

The main point is that the information that a man is a savage is not sufficient evidence to conclude that it does not speak. It is only evidence that the man may not be able to speak. In other words, we do not wish to conclude ¬speaks(x) if savage(x); we simply want to prevent a conclusion speaks(x).

The **superiority relation** among rules is used to define priorities among rules, i.e., where one rule may override the conclusion of another rule. For example, given the defeasible rules r: man (x) ⇒ speaks (x)

r´: mute (x) ⇒ ¬speaks (x)

Because these rules contradict one another, no conclusive decision can be made about whether a man who is mute can speak. But if we introduce a superiority relation with r´ > r, with the intended meaning that r´ is strictly stronger than r, then we can indeed conclude that the man cannot speak.

Notice that a cycle in the superiority relation is counterintuitive. In the above example, it makes no sense to have both r > r´ and r´ > r. Also, in Defeasible Logic, priorities are local in the following sense: two rules are considered to be competing with one another only if they have complementary heads. Thus, since the superiority relation is used to resolve conflicts among competing rules, it is only used to compare rules with complementary heads; the information r > r´ for rules r, r´ without complementary heads may be part of the superiority relation, but has no effect on the proof theory.

1.2.7.1. DefLog

The concepts presented above were applied in [Verheij 2003] to a “sentence-based” (as opposed to “argument-based”) logic for defeasible reasoning, called DefLog[4]. Verheij assumes a logical language with just two connectives, a unary connective × which informally stands for ‘it is defeated that’ and a binary connective ⇝ for expressing defeasible conditionals. He then assumes a single inference scheme for this language: *modus ponens* for ⇝:

$a \rightsquigarrow b$

a

Therefore, b

The central definition of Deflog is the notion of the dialectical interpretation of a theory, described in [Verheij 2002] as a variant of the extensions of default theories of [Reiter 1980], the models of logic programming of [Gelfond and Lifschitz 1988], the extensions of argumentation frameworks of Dung (see 1.3.2.1.), and the extensions of assumption-based frameworks of [Bondarenko et al. 1997]. A theory is any set of sentences, and when it is dialectically interpreted, all the sentences in the theory are evaluated either as justified or as defeated. This is in contrast with the interpretation of theories in standard logic, where all sentences in an interpreted theory are assigned the same positive value when the theory is given a model.

A dialectical interpretation of the theory consists in the assignment of the values justified or defeated to the sentences in a theory. Two properties are required: first, it must be conflict-free; second, it must attack all sentences in the defeated part. Following are some basic definitions:

[4] Please note that the presentation of Deflog uses notions of argumentation theory. Refer to 1.3.3.3. for concepts such as modus ponens, justified arguments and attack relations

- Let T be a set of sentences and *a* a sentence. Then T supports *a* when *a* is in T or follows from T by the repeated application of $\rightsquigarrow$-modus ponens. T attacks *a* when T supports × *a*;

- Let T be a set of sentences. Then T is conflict-free when there is no sentence *a* that is both supported and attacked by T;

- Let ∑ be a set of sentences, and let *J* and *D* be a partition of ∑, i.e. subsets of ∑ that have no elements in common and that have ∑ as their union. Then (*J*, *D*) dialectically interprets the theory ∑ when *J* is conflict-free and attacks all sentences in *D*. The sentences in *J* are the (actually) justified assumptions of the theory ∑, the sentences in *D* are the (actually) defeated assumptions. The sentences in ∑ are the theory's (prima facie justified) assumptions;

- Let ∑ be a set of sentences, and let (*J*, *D*) dialectically interpret the theory ∑. Then (*Supp J*, *Att J*) is a dialectical interpretation or extension of the theory ∑. Here *Supp J* denotes the set of sentences supported by *J*, and *Att J* the set of sentences attacked by *J*. The sentences in *Supp J* are the justified statements of the dialectical interpretation, the sentences in *Att J* the defeated statements.

1.2.7.2. The concept of reinstatement

Consider the following set of (prima facie justified) assumptions:

p

q

q $\rightsquigarrow$ ×p

It expresses that the prima facie justified assumption that *q* attacks the prima facie justified assumption *p*. The theory has one dialectical interpretation: the assumptions *q* and *q* $\rightsquigarrow$ ×*p* are actually justified, and *p* is defeated. There is one other interpreted sentence, namely ×*p*, that is justified. Consider the following set of assumptions:

p

q

q $\rightsquigarrow$ ×p

r

r $\rightsquigarrow$ × q

The attack of *q* by the prima facie justified assumption *r* has been added to the assumptions of the previous example. There is one dialectical interpretation. In it, the assumptions *p, q* $\rightsquigarrow$ ×*p, r* and *r* $\rightsquigarrow$ × *q* are actually justified, and *q* defeated. There is one other interpreted sentence, namely ×*q*, that is justified.

This example of attack and counterattack shows the phenomenon of reinstatement (presented among others by [Caminada 2006]), that is strictly related to Dung's argumentation framework (see 1.3.2.1.) and typical for defeasible reasoning: an assumption that is defeated can become justified when additional information enters into the knowledge base.

1.2.8. The relationship between Logics and Legal Reasoning

Since Leibniz proposed to apply the logic approach to law in [Leibniz 1974], with his famous foresight of men discussing moral issues through mere calculus (see 1.1.1.) many lawyers were fascinated by the idea of deriving a judicial decision from legally binding sources just like the logical deduction of a conclusion is derived from a set of axioms. Reducing judicial reasoning to logical deduction would in fact ensure certainty and testability, therefore limiting the arbitrariness of human decisions.

However, as [Sartor 2005] puts it, the relation between law and logics has not always been a peaceful cooperation. More specifically, legal experience may appear to the logician (and more generally to the scientifically trained mind) as dominated by rhetorical forms, blurry concepts, and unreasoned reliance on validity intended as a result of acts of empowering authorities. On the other hand, a lawyer may find logic trivial and barren: Its complex and technical mechanisms just make explicit what is already known, without offering any substantial help to the complex task of constructing legal solutions to new issues.

1.2.8.1. The set of oppositions by G. Sartor

Starting from the end of the fifties, many contributions such as [Perelman and Olbrechts-Tyteca 1969] tried to compare legal reasoning and the tradition of rhetoric and argumentation. Legal reasoning often defined itself according to the argumentative technique which [Perelman and Olbrechts-Tyteca 1969] call dissociation, identifying the conflict between legal reasoning and formal logic in the following set of oppositions enumerated by [Sartor 2005] in the fifth volume of the *Treatise of Legal Philosophy and General Jurisprudence*:

- Abstraction / concreteness:
    - (language)
        - Logics are *abstract*: they only take into consideration the elements which have been introduced in their languages;
        - Legal reasoning is *concrete*: it preserves the richness of natural language in order to capture the content of legal issues;
    - (context)
        - Logics are *abstract*: they rely on a formalization of the network of beliefs and attitudes underlying each piece of information;
        - Legal reasoning is *concrete*: it takes into account the social and linguistic background of the audience;
- Closeness / openness:

- (knowledge base)
  - Logics are *closed*: they assume a fixed context of knowledge;
  - Legal reasoning is *open*: its assumptions are freely chosen from general ideas, maxims or principles;
- (reasoning)
  - Logics are *closed*: they necessarily derive all valid consequences of a consistent set of premises;
  - Legal reasoning is *open*: it allows for alternative principles to be dialectically harmonized without necessarily giving birth to inconsistencies;
- (world assumption)
  - Logics are *closed*: they assume the legal system to be complete;
  - Legal reasoning is *open*: it assumes the incompleteness of any set of sources and it relies on various inferences for filling the gaps;

– Rigidity / flexibility:
  - (assumptions)
    - Logics are *rigid*: they do not distinguish between the status of its premises: all axioms are assumed to be true facts, and their conflict determines an insoluble contradiction;
    - Legal reasoning is *flexible*: it admits different types of premises, and in particular it distinguishes facts from presumptions, where the latter must be accepted only in so far as there is no evidence to the contrary;
  - (argument weight)
    - Logics are *rigid*: they find inferences to be either fully correct or fully incorrect;
    - Legal reasoning is *flexible*, since arguments may have different degrees of strength;

– Formality / materiality.
  - Logics are *formal*: they evaluates inferences only on the basis of their syntactic structure;
  - Legal reasoning is *material*: the acceptability of a legal argument depends not only on the form, but also on the substantive value of that argument (that is, on

the intrinsic goodness of its premises and conclusions or on their correspondence with social beliefs and attitudes);

- Statics / dynamics.
    - Logics are *static*: they are concerned with the eternal relationships between truths;
    - Legal reasoning is *dynamic*: it develops in time according to procedural constraints;
- Monologue / dialogue.
    - Logics are *monological*: its inferences always move from a unique consistent pool of premises;
    - Legal reasoning is *dialogical*: it consists in confronting different points of view;
- Impersonality / personality.
    - Logics are *impersonal*: its inferences are supposed be convincing for every rational being;
    - Legal reasoning is *personal*: it has to convince each particular audience, by appealing to the premises and inferences that would be acceptable to it;
- Positive / negative reasoning.
    - Logics are a *positive* form of reasoning: they support a conclusion by showing that it is derivable from certain premises;
    - Legal reasoning emphasises *negative* reasoning: it usually supports an argument by undermining its competitors;
- Objectivity / evaluation.
    - Logic are *objective*, since they excludes choices: either there is a contradiction, so that the whole axiomatic base collapses, or there is consistency, so that no choice is required;
    - Legal reasoning is *subjective* (evaluative): it admits conflicting theses and supports the evaluative choices required to adjudicate between them;
- Constraint / persuasion.
    - Logic is *constraining*: it is not possible, for any rational being, to refuse assent to a logically valid inference;
    - Legal reasoning is *persuasive*: it leads its addressees towards a certain conclusion without forcing its acceptation.

In the 1970’s the dispute reached a stalemate. In fact, the *formal* approach — even if extended beyond first-order predicate logic thanks to the equipment of modal and deontic logics — was incapable of giving a reasonably wide account of legal reasoning: legal logic was in fact effective only for limited fragments of law text, where a logical formalization was possible. The logical application of the law therefore required a radical change—with regard to the usual legal practice—both in the formulation of legal knowledge and in the methods of legal inference. On the other hand, the informal approach—even if extended beyond general rhetorical moves, into specific aspects of legal reasoning—was more successful in pointing to problems than in finding convenient solutions. Its refusal of any formal tools impeded the development of a precise definition of the forms of legal reasoning.

In the last years, the hostile opposition between logic and argumentation has been largely overcome through fruitful interaction between formal logic and legal reasoning. This started in legal theory, where many scholars have approached legal reasoning adopting an analytical perspective. Some authors such as [Alchourron and Bulygin 1971, Tammelo 1978] have applied in innovative ways classical and deontic logic to the analysis of legal reasoning, and others such as [Tarello 1980, Alexy 1989, Taruffo 1998] have embedded deductive reasoning within broader accounts of legal argumentation. Analytical legal theory has indeed offered a common language for logic and jurisprudence. More recently, in the field of Artificial Intelligence and Law (See 1.4.), some new logical accounts of legal reasoning have been provided [Gardner 1987, Gordon 1995, Prakken 1997, Branting 2000]. These accounts preserve the preciseness of deductive logic, but try to further extend the formal analysis of legal reasoning. These recent results (presented in the following sections) seem to indicate that the conflict between logic and informal accounts of legal reasoning could after all end in a harmonious cooperation.

1.2.8.2. Enthymemes

A familiar problem in the evaluation of non-formal legal rules (rules whose logical structure is not explicit) and non-formal legal arguments is that they are very often enthymematic.

An enthymeme is any rule or argument whose logical form is not explicitly clear from its original mode of presentation (for example, in a judicial sentence or a lawyer's act). Enthymemes have been investigated, among others, by [Brewer 1996] and [Walton 2001]. In general, two types of enthymeme can be identified: the *rule-enthymeme* and the *argument-enthymeme* (see 1.3.1.3. for a description of the latter). It is however unusual to find enthymemicity attributed to rules in the literature. It is much more easy to find the property of enthymemicity attributed to arguments, and enthymemicity is indeed an important property of many arguments, especially legal arguments.

Here is an example of a **rule-enthymeme** from [Brewer 1996]:

> Any person who knowingly transports stolen property over state lines is guilty of a felony

The point here is to determine whether R equivalent to

> Any person who transports over state lines property that he knows is stolen is guilty of a felony

or to

> Any person who knows that he is transporting over state lines property that is stolen is guilty of a felony

Note that under R2 a person can be guilty of the felony even if he doesn't know that the property is stolen, whereas under R1 he is not guilty unless he knows that it's stolen (indeed, under R1 he need not even know that the stolen property has been taken over state lines). This is a particularly obvious example of a rule-enthymeme, because R itself is ambiguous.

1.3. Argumentation Theory

1.3.1. Introduction

Logics constitutes a huge efforts towards the formalization of abstract phenomena that everybody can witness, such as the connection between truth and its objects. This discipline, however, despite having developed in several directions and covering a very widespread and detailed set of mechanics, always remains on the formal, mathematical side of the representation. It is the discipline of argumentation that cares for the adaptation of the logic schemes to real phenomena, by distinguishing among different logical forms and guiding the evolution of logics while being guided by it, in a continuous comparison of abstract mathematical models and qualitative reasoning.

The concept of logically valid inference is seen as 'foolproof' reasoning: an argument is deductively valid if the truth of its premises guarantees the truth of its conclusion. In other words, if one accepts all premises of a deductively valid argument, then one also has to accept its conclusion, no matter what. However, real life arguments are often not foolproof in this sense but merely make their conclusion plausible when their premises are true. For example, if we are told that Hans and Ingrid are married and that Hans lives in Berlin, we conclude that Ingrid will live in Berlin as well, since we know that usually married people live together. Such arguments can be overturned by counterarguments: for example, if we are told that Ingrid works at the foreign offices of her company in Rome for two years, we have to retract our previous conclusion that she lives in Berlin. As long as such counterarguments are not available, the conclusions of our fallible arguments hold. As seen in 1.2.6., logics for defeasible reasoning (partly inspired by earlier developments in philosophy and argumentation theory) were developed in the last decades to manage such kind of reasoning. At first sight it might be thought that patterns of defeasible reasoning are a matter of applying probability theory. However, many patterns of defeasible reasoning cannot be analyzed in a probabilistic way. In the legal domain this is particularly clear: while reasoning about the facts can be regarded as probabilistic – at least in principle, reasoning about normative issues clearly is of a different nature. Moreover, even in matters of evidence reliable numbers are usually not available so that the reasoning has to be qualitative.

1.3.1.1. Different kinds of reasoning

The first concepts to be introduced in order to delve into a presentation of Argumentation theory are the different forms of valid reasoning (*deductive, inductive, abductive, analogical*) and the distinction of normal premises from assumption and exception.

Deductive reasoning is one of the two basic forms of valid reasoning. It begins with a general hypothesis from which it generates a specific conclusion. The basic idea of deductive reasoning is that if something is true of a class of things in general, this truth applies to all members of that class. One of the keys for correct deductive reasoning, then, is to be able to properly identify the members of the class. If the generalization is wrong, though, the specific conclusion can be logical and valid but still can be incorrect. For example, a generalization might be "All bulls have horns". The logical conclusion of a specific instance would then be, "That is a bull, so it has a horn." This is a valid deduction, the truth of which depends however on whether the observed animal is, indeed, a bull.

Inductive reasoning is a method of drawing a probable conclusion from an emerging pattern of data. When data show a large scale of regularity, an analyst can logically predict that those patterns will continue to repeat. This inference, commonly known as generalization, can produce scientific deductions so probable that they are widely accepted as fact. Any theory involving generalization, however, can be disproved by one instance of inconsistency. In medicine, this type of inductive reasoning can be a very powerful diagnostic tool. As a specific illness often presents with a particular list of symptoms, it is reasonable to presume that a patient who exhibits those indicators also has that malady.

Abductive reasoning is a form of reasoning based on the formation and evaluation of hypotheses using the best available information. It starts with the observation of a phenomenon for which one does not have an immediate, clear explanation. One can then use this form of reasoning to develop an explanation that is sufficient to describe the observed phenomenon, though it must be noted that, without further testing, this explanation is only sufficient, not necessarily accurate. Abductive reasoning is similar to inductive reasoning, but only involves developing a guess based on what limited data is available at a given time, before detailed testing and rigorous observation.

Analogical reasoning is a method of processing information that compares the similarities between new and understood concepts and then uses those similarities to gain understanding of the new concept. It is based on the brain’s ability to form patterns by association. Similarly to abductive reasoning, analogical reasoning is a form of inductive reasoning because it aims to identify what is likely to be true, rather than deductively proving something as fact. The reasoning process begins by a person determining the target idea to be evaluated. It is then compared to a similar idea that is already well-understood. Specific qualities that belong to this known idea are then chosen, and related to the qualities in the target idea to find similarities. For example, trying to catch a valuable baseball can be an analogy to pursuing a fox because they are both efforts put towards an expected gain.

1.3.1.2. Different kinds of premises

A basic argument may have multiple conclusion and multiple premises. Defeasibility tells how arguments do not always support conclusions in the same way, but it is necessary to introduce a distinction also between premises: aside from *ordinary premises*, which support the validity of arguments only as far as they are accepted, there are

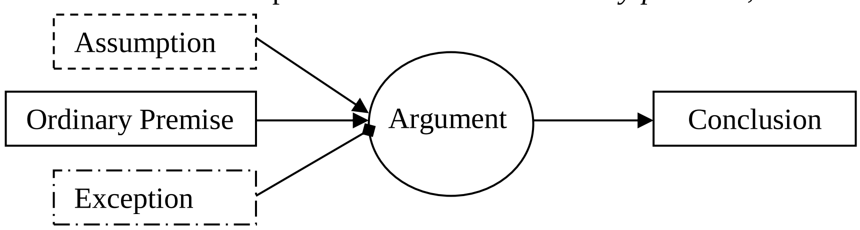


Fig. 1.11: basic argument structure.

assumptions and *exceptions*.

Assumptions behave like ordinary premises, with the exception that they do not prevent the validity of arguments unless they are rejected: their default value is favorable to the validity of the argument as if the premise was accepted. On the contrary, **exceptions** are premise which, if accepted, would render the argument invalid. In order not to prevent the validity of the argument, they must be rejected. However, their default value is favorable to the argument as if the exception was rejected: they do not prevent the validity of arguments until they are accepted. In other words, their negation is assumed.

This definition of assumption and exception follows the conception of *assumption-based frameworks* (see [Bondarenko et al. 1997]), and is different from that proposed by [Gordon et al. 2007] where the assumption is not equivalent to the negation of an exception (see 4.2.2.2.)

1.3.1.3. Argument enthymemes

Here is an example of an *argument enthymeme*. Suppose a judge writes in an opinion resolving a contracts dispute:

> The general clause is not knowable, so it is invalid

This might be represented as a premise *p1* that provides inferential warrant for conclusion *h*:

> p1: The general clause is not knowable.

therefore,

> h: The general clause is invalid.

Taken as literally quoted, the argument seems not to be a valid deductive argument. By its literal terms the argument provides no reason to believe that every general clause that is not knowable is invalid. Without more, it is conceivable that some types of general clauses are valid.

In fact the argument, properly interpreted, is not deductively invalid. If we believe that the judge offering this argument from *p1* to *h* was relying on the unstated but assumed premise, 'All general clauses which are not knowable are invalid', then we would conclude that the best way to interpret the judge's argument is as follows:

> p0: All general clauses which are not knowable are invalid'

> p1: The general clause is not knowable.

therefore,

> h: The general clause is invalid.

This is a valid deductive argument. In this example, we conclude that the true logical form of the argument (that premises *p0* and *p1* provide inferential warrant for *h*) was not explicitly clear from the way in which it was originally presented; at first glance it seemed like the argument was that *p1* by itself provided the inferential warrant for *h*. But, on second analysis, we might judge that the argument is an enthymeme, an argument, as defined above, whose logical form is not explicitly clear from its original mode of presentation but whose proper logical can be distinguished only through a more formal representation.

Note that it is necessary to pay attention to the circumstances under which interpreters are warranted in treating arguments as enthymemes: in fact, every argument could be interpreted as a valid deductive argument, but surely not every argument is a valid deductive argument or indeed is a deductive argument at all – some are inductive, some analogical, some abductive. For example:

We provide the example for that by representing Aristotle´s syllogism, presented in section 1.2.1 when first introducing Logic, and which Philosophers have long offered as the paradigm of a valid deductive inference:

p1: All men are mortal.

p2: Socrates is a man.

Therefore,

h: Socrates is mortal.

This does indeed seem to be a valid deductive inference, for it does seem that in any possible world in which p1 and p2 are true h must also be true. But consider what kind of justification there could be for the first premise, 'All men are mortal.' Surely it rests on an inductive generalization (highly confirmed, that's for sure). Might one not fairly represent the "Socrates syllogism" not as a deductively valid inference but as an inductive specification, that is, an application of an inductive generalization to an individual, where the major premise is not assumed or known to be a true universal generalization, which inductive generalizations are incapable of producing? What criteria should be used? When relying on logics of legal argument such questions are fundamental.

1.3.2. Reasoning with defeasible arguments

As [Prakken and Sartor 2009] explains, reasoning with defeasible arguments consists of constructing arguments, attacking these arguments with counterarguments, and adjudicating between conflicting arguments on grounds that are appropriate to the conflict at hand. In deductive reasoning, arguments must instantiate inference schemes (called 'argument schemes' in the present work, following [Walton 2005] – see 4.2.1.1.) which represent fool-proof reasoning: in the presence of defeasible arguments, however, deductive logic turns out to be the special case of argument scheme that can only be attacked on their premises.

Consider the following example. According to Article 1341 of Italian Civil Code, oppressive clauses are not efficacious unless specifically signed. Let's suppose Clause *x* is argued to be inefficacious since it is oppressive and not specifically signed. Then in standard propositional logic this argument would be:

Argument A:

The clause is oppressive & $\neg$ The clause is signed $\rightarrow$ $\neg$ The clause is efficacious

Clause x is oppressive

$\neg$ Clause x is signed

Therefore, $\neg$ Clause x is efficacious

This argument is deductively valid, since it instantiates the deductively valid argument scheme of *modus ponens*:

Modus Ponens Scheme:

$P \rightarrow Q$

P

Therefore, Q

This scheme is deductively valid since it is impossible to accept all its premises but still deny its conclusion: the truth of its premises guarantees the truth of its conclusion.

However the deductive validity of argument A does not mean that its conclusion is accepted, as the deductive argument can still be challenged on its premises. According to Decision 1248/2003 of the Italian Cassation Court, clauses contained in statutes known to parties are considered as signed. The following deductive argument against the premise "$\neg$The clause is signed" can be constructed, in two steps. First application of Decision 3848 results in the conclusion that the clause is signed:

Argument B:

The clause is contained in a statute $\wedge$ The statute is known $\rightarrow$ The clause is signed

Clause x is contained in a Statute y

Statute y is known

Therefore, Clause x is signed

This conclusion can then be used to attack the third premise of argument A:

Argument B (continued):

Clause x is signed

Clause x is signed $\rightarrow$ Clause x is efficacious

Therefore, Clause x is efficacious

It is necessary to choose whether to accept the premise "$\neg$ law provides otherwise" of argument A or to give it up and accept the conclusion of counterargument B. Clearly the phrase "unless the clause is specifically signed" of Article 1341 of Italian civil code is meant to express that any case in which the clause is specifically signed represents an exception to Article 1341. Since argument B is based on such an exception, the premise of A is rejected and the counterargument is accepted. In this case argument B not just *attacks*, but also *defeats* argument A[5].

However, not all attacks are a matter of statutory exceptions. In the example, argument B could be attacked by saying that the statute was not known to parties. This gives rise to an argument attacking the third premise of argument B (before the continuation):

Argument C:

The statute is known $\rightarrow$ The clause is signed

$\neg$Statute *y* is known

Therefore, $\neg$Clause *x* is signed

This time a genuine conflict has arisen, between the claim that the statute is known to parties (the third premise of argument B) and the claim that the status is not known to parties (the second premise of argument C). Please note that if one accepts all premises of argument C, then one must also accept its conclusion, since argument C instantiates the deductively valid scheme of modus ponens. And if one accepts argument C's conclusion, one must, of course, reject the third premise of argument B. In the latter case argument C not only *attacks* but also *defeats* argument B. So far all three arguments are deductively valid but argument A is defeated by argument B on its third premise while argument B is in turn defeated by argument C on its third premise. This implies that it is rational to accept the conclusions of arguments A and C: even though A is defeated by B, it is *defended* by C, which defeats A's only defeater.

In order to determine what to believe or accept in the face of a body of conflicting arguments it does not suffice to make a choice between two arguments that directly conflict with each other: it is also necessary to look at how arguments can be defended by other arguments. In the example it is intuitively obvious that C defends A and, since C is not attacked by any argument, both argument A and argument C (and their conclusions) are acceptable. However, it is easy to imagine more complex examples where intuitions fall short. For instance, another argument D could be constructed such that C and D defeat each other, then an argument E could be constructed that defeats D but is defeated by A, and so on: which arguments can now be accepted and which should be rejected?

Here intuitions are out of play and a calculus, or an argumentation logic, is needed. Its input will be a collection of arguments plus an assessment of which arguments defeat each other, while its output will be an assessment of the *dialectical status* of these arguments in terms of three classes:

– The **justified** arguments are those that survive all conflicts with their attackers and so can be accepted;

[5] This definition allows that two arguments defeat each other, namely, if neither argument is inferior or superior to the other. In such cases we say that the two arguments *weakly defeat* each other; otherwise (if one argument is superior to the other) we say that one argument *strictly defeats the other*.

- The **overruled** arguments are those that are attacked by a justified argument and so must be rejected;
- The **defensible** arguments are those that are involved in conflicts that cannot be resolved.

Furthermore, a statement is justified if it has a justified argument, it is overruled if all arguments for it are overruled, and it is defensible if it has a defensible argument but no justified arguments.

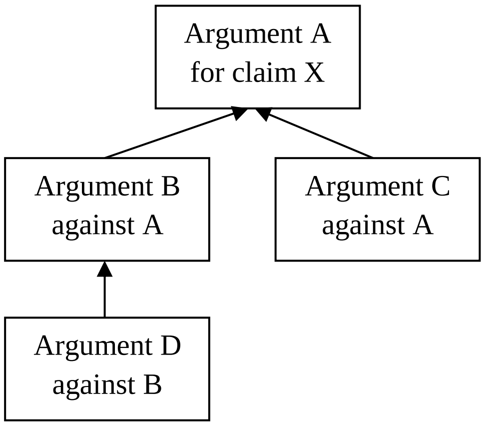


Fig. 1.12 – A dialectical tree

1.3.2.1. Dungean semantics and the problem of new information

The logic of argumentation relies on a dialogical rather than a monologue-like structure. The proponent starts with the argument that he wants to justify and then the turn shifts to the opponent, who must provide all its defeating counterarguments. For each of these defeating arguments the proponent must then construct one strict defeater (it has to be a strict defeater since the proponent must prove his argument justified). This process is repeated as long as it takes: at each of her turns, the opponent constructs all mutual and all strict defeaters of the proponent's previous arguments, while at each of his turns, the proponent constructs a strict defeater for each of the opponent's previous arguments, and so on. The initial argument is justified if the proponent can eventually make the opponent run out of moves (counterarguments). This process can be visualized as shown in Figure 1.12.

Note that if an argument is justified this does not mean that the proponent will in fact win the game: he could make the wrong choice at some point. All that it means is that the proponent will win if he plays optimally. In terms of game theory [Rahwan and Larson 2009], an argument is justified if the proponent has a so-called *winning strategy* in a game that starts with the argument. In fact, there is a simple way to verify whether the proponent has a winning strategy. The idea is to label all arguments in the tree as *in* or *out* according to the following definition of [Dung 1995]:

> An argument is *in* if and only if all its defeating counterarguments are *out*
>
> An argument is *out* if and only if it has a defeating counterargument that is *in*

Suppose now that new information becomes available that gives rise to a strictly defeating counterargument. Then the situation is as in Figure 1.13.

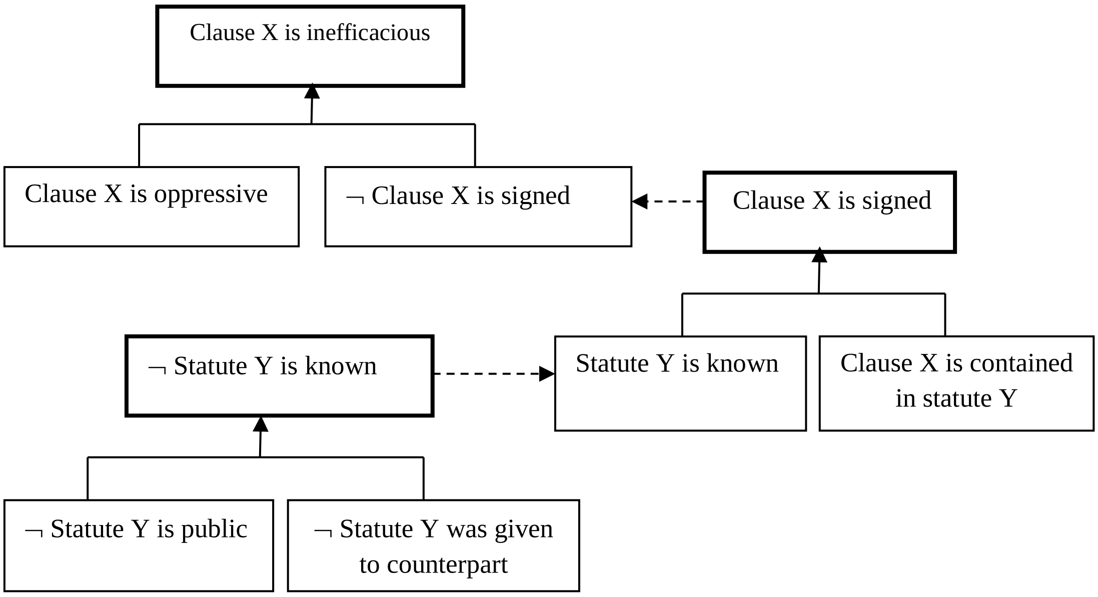


Fig. 1.14 - Three arguments and their defeat relations

Now it is the proponent who has a winning strategy. This illustrates that when new information becomes available from which new arguments can be constructed, the dialectical status of arguments may change. The phenomenon is known as reinstatement (see 1.2.7.1.).

It should be noted that each argument appearing as a box in these trees has an internal structure. In the simplest case it just has a set of premises and a conclusion, but when the argument combines several inferences, it has the structure of an inference tree similar to that of standard logic. This is illustrated by Figure 1.14, which displays all three arguments of the example, plus their defeat relations (solid lines represent inferences while dashed lines stand for defeat relations):

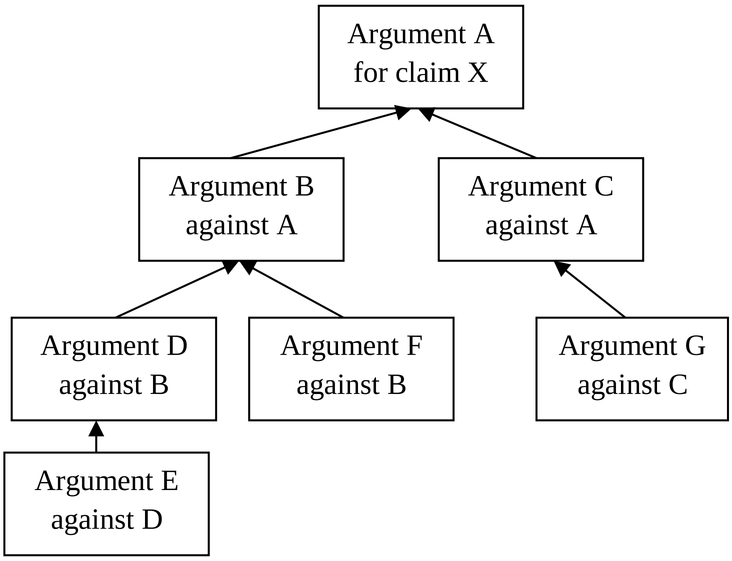


Fig. 1.13 – An extended dialectical tree.

1.3.2.2. Defeasible rules and generalizations

Giving a close look at arguments as they are constructed and attacked in practice, it is clear that they can often be attacked even if all their premises are accepted. In fact, many factual generalizations that are used in daily life and also many that are used in in legal proof are of presumptive or *defeasible* nature. At the beginning of this section the example "usually married people live where their spouses live" was given, which can have exceptions, for instance, when a spouse temporarily works abroad. Not only factual generalizations are defeasible but also, for example, interpretation rules or reasons for action. An example of a defeasible interpretation rule in contract law is "a statement 'I accept ...' is an acceptance", but an exception was "a statement 'I accept' followed by terms that do not match the terms of the offer is not an acceptance" (see [Gardner 1987]). It has even been argued that legal rules are also defeasible, since there can always be unforeseen cases in which a rule should be set aside because of higher principles or unwanted consequences.

As [Prakken 1997] noted, these features do not escape a logical analysis, if recent developments in logic and Artificial Intelligence on so-called nonmonotonic reasoning and defeasible argumentation are used, and if logic is regarded as a tool in (rather than as a model of) legal argumentation. An important point is that the application of defeasible generalization cannot be regarded as an instance of the modus ponens argument scheme, since P → Q means that always when P is true then Q is true, and this is not the same as saying that usually when P is true then Q is true. So a new argument scheme is needed, namely the defeasible modus *ponens*:

Defeasible modus ponens:

If *P* then usually *Q*

P

Therefore (presumably) *Q*

In [Prakken 2010], this scheme is used as the basic building block for reasoning about rules, together with the underlying *rule validity scheme* inspired from the work of [Sartor 2008] and [Bex 2009]:

Rule validity scheme:

Rule *r* is valid

Therefore

If *P* then usually *Q*

The previous section explained how argumentation logic systematizes this process of testing an argument in light of all possible counterarguments, but an argument can be attacked not only on its premises, but also on its conclusion.

In fact, a conclusion of an argument can be attacked in a stronger and a weaker way. The strong way is to build an argument with the opposite conclusion, as done above with argument D. Such a conclusion-to-conclusion attack is called a *rebutting attack* (or rebuttal). A rebutting counterargument may attack the final conclusion of its target but it may also attack an intermediate conclusion. However, sometimes an argument can be attacked in a weaker way, namely, by saying that the premises, even if true, do not support their conclusion in the case at hand, because the case at hand is an exceptional case. This weaker form of attack is often called *undercutting attack* (or undercutter). Undercutting counterarguments do not attack a premise or the conclusion of their target but instead deny that the scheme on which it is based can be applied to the case at hand. Obviously, such a denial does not make sense for deductive argument schemes. See 1.3.4.2. for an example of undercutting attack.

1.3.2.3. Argumentation Schemes

When looking at defeasible rules or generalisations, it can be noted that often they are not just specific statements about the world but conform to certain reasoning patterns. It is important to note that arguments based on such patterns speak about states of affairs in general: unlike specific generalisations like 'summer in Italy is usually very hot' or 'fleeing from a crime scene typically indicates consciousness of guilt' they express general ways of obtaining knowledge from certain information. For these reasons it is natural to regard such patterns not as conditional premises of a defeasible modus ponens argument but as additional presumptive (or defeasible) argument schemes. For instance, the defeasible argument scheme for witness statements can be given the following form:

Argument scheme from Witness Testimony:

Person W says that P

Person W was in a position to observe P

Therefore (presumably), P

The use of presumptive argument schemes in an argument grants new expressivity to the two new ways of attack presented in the last section: rebutting and undercutting attacks. For example, an application of the scheme from witness testimony can be rebutted by an application of the same scheme to a contradicting witness. Undercutting attacks are instead based on the idea that a presumptive argument scheme has typical exceptional circumstances in which it does not apply. For example, a witness testimony is typically criticised on the witnesses' truthfulness or the functioning of his memory or senses. In general, then, each argument scheme comes with a set of *critical questions* which, when answered negatively, give rise to undercutting counterarguments (or sometimes to rebutting counterarguments). For example, the Witness Testimony Scheme is often given the following critical questions:

Critical questions to the Argument Scheme from Witness Testimony

W1: Is the witness truthful?

W2: Did the senses of the witness function properly?

W3: Does the memory of the witness function properly?

Please notice that while some schemes can arguably be used in any domain, other schemes are domain-dependent. For instance, it has been argued that in legal contexts the Argument Scheme for Witness Testimony has different critical questions than in ordinary commonsense reasoning. See Chapter 3 for a complete presentation of Walton's theory on Argumentation Schemes, which constitutes the backbone of the Carneades Argumentation System.

1.3.3. History of Diagramming in Logic

This section contains a brief description of how argument diagramming has evolved as a tool for the critical analysis of everyday argumentation from the nineteenth and through the twentieth century, mostly following [Reed et al., 2007].

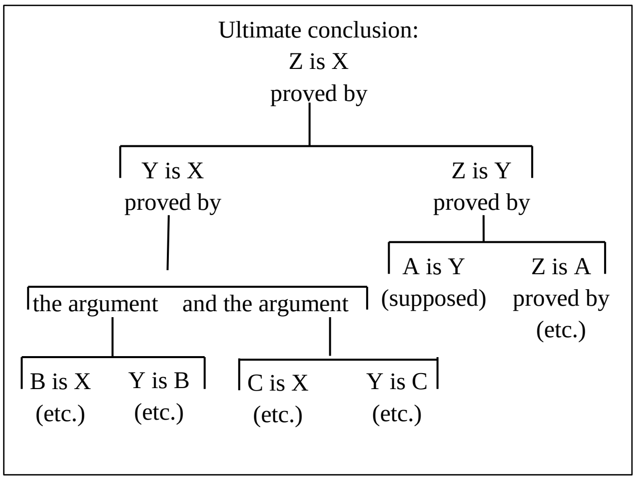


Fig.1.15 – Whately's diagramming [Whately, 1836, p. 422]

The 19th century saw not only a revival of interest in formal logic, but also an accompanying interest in representations of formal systems. Venn and Euler diagrams for syllogistic reasoning [Venn 1880], and other visual techniques meant to represent logical reasoning of one sort or another could be cited.

1.3.3.1. R. Whately: The invention of argument diagrams

The first example of diagrams used to illustrate argumentative processes may be traced back to Richard Whately in 1836. Whately, an English logician and Archbishop of Dublin, in Appendix III of his textbook Elements of Logic [Whately 1836], entitled 'Praxis of Logical Analysis', described a method of argument analysis. He described it as a method of taking 'any train of argument that may be presented to us', and reducing it to a form in which logical rules can be applied to it. The method consists in trying to figure out what the conclusion of the argument is supposed to be, and then tracing the reasoning backward, to find out the grounds of that assertion. The process can then be repeated, searching for further grounds for these premises. The outcome is what Whately described as the construction of a 'chain of arguments', a process he represented by a diagram (Figure 1.15). This diagram has many of the basic characteristics of the modern argument diagram. Statements are represented as the nodes, joined by lines to make up a tree or graph structure. The structure represents a chain of argumentation with an ultimate conclusion at one end.

Whately represents an isolated case in the 19th century, and argument diagramming only resurfaces in the proliferation of logic texts using diagramming in the 20th century.

1.3.3.2. J. H. Wigmore: increasing complexity

If Whately is considered the pioneer of diagramming arguments in the logical field, John Henry Wigmore was the first to visually represent, in [Wigmore 1913], complex diagrams to represent proof-hypothesis in legal matters. His schemes were disregarded after his death, but his idea of organizing evidential arguments has been recently reconsidered and developed in [Schum et al. 2005]. He can be regarded as the initiator of the current of the study of using diagramming to map facts and inferential links in a body of evidence in a case at trial in law.

The chart in Figure 1.16 represents evidence in a case from [Wigmore 1931]. In this diagram, Wigmore indicated the statement "Y died of poison" as being the *ultimate probandum*, at least of this part of the evidential argument, Circle 7 is an *interim probandum*, and the line connecting 7 with the ultimate probandum means "provisional probative force given to the evidence". The other kind of inference is the type representing strong probative force, connecting, in this example, 8, 9, 10 with 7. The focus of Wigmore's interest is in demonstrating the acceptability of the hypothesis given the factual evidence. The direction, consequently, is

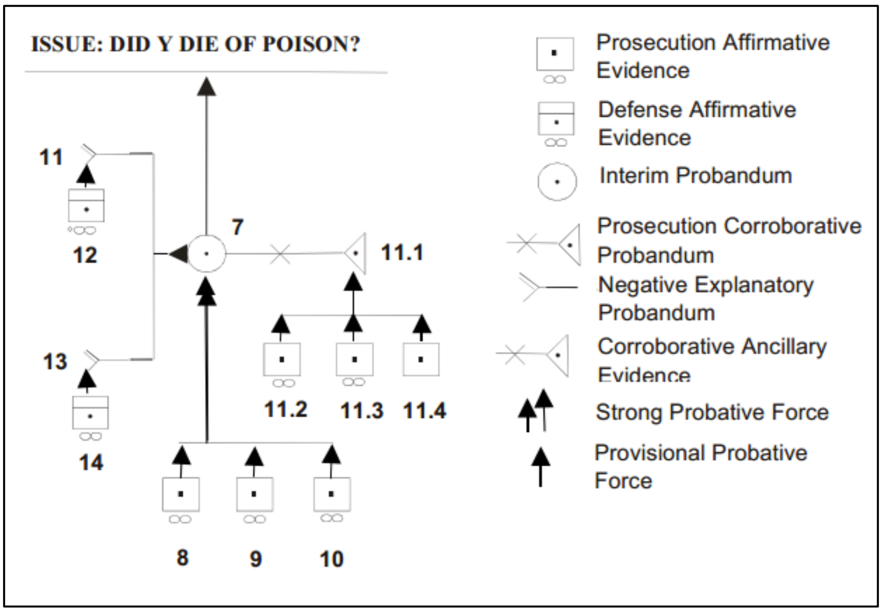


Fig. 1.16 – Wigmore diagram example from [Walton et al. 2007]. Key list: 7. Y died, being apparently in health, within three hours after the drink of whiskey; 8-10 Y's Wife and the Northingtons witness to 7; 11 Y might have died by colic from which he had often suffered; 11.1 colic would not have had as symptoms the leg cramps and teeth-clenching, only strychnine could produce these; 11.2 Y's wife and the Northingtons witness to Y's cramps and teeth-clenching; 11.3 expert witness to significance of symptoms; 11.4 no testimony as to strychnine traces in the body by post mortem; 12 anon witness to his former attacks; 13 Y might have died from the former injury to his side; 14 anon witness to that injury.

upward, from evidence to hypothesis. The arrow direction indicates the kind of hypothesis-evaluation approach Wigmore developed in his theory. It proceeds from the evidence to the hypothesis, the latter being proved or disproved by the evidence.

Another interesting feature of Wigmore diagrams is the notion of complex inference. The probandum is supported by evidence, which is in turn supported by other evidence. The whole process of justifying the hypothesis is constituted by a complex argumentation where facts are warranted by other proofs. Evidence, in his words, is not certain, but must be supported in order to be acceptable as a conclusive proof. [Macagno et al. 2007] notes that by utilizing complex inferences Wigmore introduced what now is being analysed by the term *inference networks*: nets of links between nodes, influencing each other's probabilities.

From these characteristics follows the third main feature of Wigmore's charts: the *conditional dependency* of arguments. Arguments are related to each other by dependency links, and their probability is influenced by the probability of the supporting evidence. The force of the ultimate conclusion, for this reason, is the result of a complex calculus of probabilities and factual probabilities. In Wigmore's theory inferential links themselves are not deemed relevant in the consideration of the relationship evidence-conclusion. They do not need to be warranted: the calculus of probabilities is based only on proofs (nodes), not on the strength of the inference.

Finally, Wigmore, in his diagrams, introduced triangles to indicate a form of evidence distinct from the other kinds of affirmative evidence (squares). These proofs are called *ancillary* - that is, they affect the probability of the evidence: in Wigmore they are considered necessary to establish and evaluate a hypothesis about a fact. In modern theories this notion has developed through the theories of probabilities and inferences, in evidence supporting generalizations conceived by David Schum (see 1.3.3.5.).

1.3.3.3. A. Beardsley: serializing arguments

Until the 1950s the theory of argumentation was taken up wholly by the predominant interest in formal logic. The first example of argument mapping after Wigmore is from Beardsley's *Practical Logic*. In the diagram of an argument supporting the necessity of freedom in the arts, he divided the argumentative text into statements. He represented the statements as nodes, using circled numbers, and he represented the links between the premises and the conclusion as arrows joining the nodes. He drew what he defined as the 'skeletal pattern' of the argument, representing its structure. Beardsley identified different kinds of links proceeding from *reasons* (premises) to conclusion: they may back track, shift gear in the middle, run in a circle or go off in several directions [Beardsley 1950]. He defined a *serial argument* as a statement that is both conclusion and reason for a further conclusion, and formulated some important general principles of diagramming, such as the *Rule of Grouping* (if you have several reasons for a certain conclusion, they should be kept as close together as possible), or the *Rule of Direction* (if you have a serial argument, it should move in one direction, no matter which one). Beardsley diagrams are graphs meant to teach how to organize the reasons for a claim, by examining the different kinds of argument structures representing reasons supporting the claim as a conclusion. They were conceived as an aid in the detection of fallacies like arguing in a circle (*petitio principii*).

In Beardsley’s model, however, arrows link reasons and conclusions, no relevance being given to the implication itself between them. There is no theory, in other words, of inference distinguished from logical deduction: the passage is always deemed not controversial and not subject to support and evaluation.

1.3.3.4. S. Toulmin: weighting the links between arguments

The main revolution in the theory of argumentation was carried out by Stephen Toulmin's “*The Uses of Argument*”, published in 1958. Toulmin can be considered the first in the theory of argumentation to take into consideration the defeasible generalization used as the step between the premise and the conclusion of an argument. To analyse this step, Toulmin introduced the concept of *warrant*, which he saw as a hypothetical statement that can be subject to defeat in some cases acting as a bridge or link between the two poles. He compared warrants with questions of law as opposed to questions of fact. For example, the fact that a man was born in Bermuda leads to the conclusion that presumably he is British because there is a law that warrants that inference [Toulmin 1958]. Toulmin also introduced the qualifier representing the degree of force of the inferential link (necessarily, probably, etc.), which makes the inference defeasible because it can fail to hold in some cases. Thus in his scheme two other factors are prominent: the rebuttal (the exceptional conditions that might defeat the conclusion) and the backing (the assurances supporting the inferential passage).

In his later work, “*An Introduction to Reasoning*”, he classified commonly used forms of argument, comparable to the ancient τόποι. Toulmin was a man well ahead of his time: during the heyday of positivism, in which only deductive and inductive reasoning of the Bayesian kind were recognized as forming rational arguments of an objective kind, Toulmin boldly set out a paradigm of rational argument that was defeasible, opening the way to the study of argumentation schemes that do not fully belong to deductive or inductive form.

1.3.3.5. M. Scriven: introducing rebuttals

Michael Scriven introduced the evaluation of the role of the premises in supporting the conclusion. He represents the counterargument in his diagrams, taking into account what Toulmin defined as rebuttal (an argument leading to a conclusion contrary to the main one), and considering it to be a legitimate and important form of argument. [Scriven 1976] distinguishes premises *pro* and *contra* by marking the former with + and the latter with -. He also indicated missing premises in his graphs, designed with an alphabetical letter instead of a number. The diagrams become more complex when the conclusion is supported by several premises, which are in their turn backed by other assumptions. They constitute, in such cases, an argument network.

1.3.3.6. D. Schum: introducing generalizations

Wigmore's ideas were further developed in a new theory on evidence in [Schum 1994], based on Bayesian probabilities and on Toulmin's analysis of inferences. The passage from evidence to a conclusion is defined as a *generalization*. Generalizations can be interpreted as forms of warrant that in some cases fall under the main categories of argumentation schemes. They allow a conclusion to proceed from premises that function as evidence, and in this perspective their function and nature covers the role of the ancient τόποι. Schum's interest is focused on the probability of the link between the nodes, and ancillary evidence acts like Toulmin's backing, strengthening or weakening the inferential step. The function of this kind of evidence is very close to the notion of critical questions in Walton's theory [Walton 1996]: they provide critical elements to evaluate the reliability of the proof. An important feature of Schum’s graphs is the inference network: the pieces of evidence may be related to each other, forming dependencies networks. This notion became extremely important after the introduction of the probabilistic calculus based on the Bayesian approach.

1.3.4. Argument diagrams in AI

As [Reed et al. 2007] puts it, there is a natural, bidirectional relationship between arguments expressed in diagrams and knowledge represented by AI systems through argumentation theory. On the one hand, argumentation theoretical structures in AI are often presented and explored using argument diagrams, with those diagrams acting as an abstraction mechanism. For this sort of presentation, internal structures of arguments are relatively unimportant, whilst the attack relationship between propositions forms a central focus of both the theory and its diagrammatic exposition [Grasso et al. 2000, Carenini and Moore 2001]. On the other hand, diagrams are also used informally to visualize and explore complex problems involving different kinds of knowledge, with these diagrams informing and framing the subsequent development of the theoretical and implemented machinery that handles such information [Crosswhite et al. 2003].

1.3.4.1. H. Rittel: an Issue-Based Information System

The first important contributions to computer-based argument diagrams were brought forward during the '60s and '70s by Horst Rittel, which – together with Werner Kunz – invented the Issue-Based Information System (IBIS). IBIS is an argumentation scheme designed to support coordination and planning of political decision processes [Rittel 1972, Kunz and Rittel 1970, Rittel and Noble 1989, Kirschner et al. 2003]. IBIS is used in issue mapping [Okada et al. 2008], an argument visualization technique related to argument mapping. It is also the basis of a facilitation technique called dialogue mapping [Conklin 2003]. Its basic structure is a treeview, a method often used in AI [Pearl 1984], and therefore it is meant to be used through a computer.

The elements of IBIS are issues that need an answer, each of which is associated with alternative positions – or possible answers. These in turn are associated with arguments which support or object to a given position. In the same way are treated new issues which come up during the processing of the initial issues.

The purpose of such systems is to widen the coverage of a problem. By encouraging a greater degree of participation, particularly in the earlier phases of the process, the designer wanted to increase the opportunity that difficulties of his proposed solution, unseen by him, will be discovered by others. Since the problem observed by a designer can always be treated as merely a symptom of another higher-level problem, the argumentative approach also increases the likelihood that someone will attempt to attack the problem from this point of view. Another desirable characteristic of the Issue-Based Information System is that it helps to make the design process “transparent”, as participants are allowed to trace back the process of decision-making.

1.3.4.2. J. Pollock: a classification of counterarguments

Perhaps one of the most influential theoretical frameworks is that of [Pollock 2002]. John Pollock focused his interest on the phenomenon that Toulmin defined as rebuttal [Toulmin 1958]. Using tree diagrams to represent reasoning, he analyzed how a conclusion can be defeated, weakened or refuted by a counterargument. In his view, a counterargument can attack the argument at which it is aimed in two ways: it can refute the conclusion itself or it can attack the inferential link between the premises and the conclusion. The first kind of refutation (closer to Toulmin's rebuttal) is defined as a *rebutting defeater*. A given proposition S concluded on the basis of a premise R is rebutted when another proposition Q is a reason for denying S. On the other hand, an *undercutting defeater* aims to undermine the inferential link between premises and the conclusion. As his leading example, Pollock considers the case of an object *x*, looking red, illuminated by red lights. The inference is from the perception to the reality of the observed phenomenon: if the object looks red, it is red. The undercutting defeater intervenes by attacking the passage between perception and reality. The fact that the object is illuminated by red lights is not a rebuttal of the conclusion however, because a red object illuminated by a red light looks red. It gives reasons, instead, for doubting that *x* wouldn't look red unless it was red: that, in other words, the premise guarantees the conclusion [Pollock 2002]. Pollock defines in [Pollock 1995] such defeaters as *reliability defeaters*, for their action works against the reliability of a reason.

Another important topic raised by Pollock concerns the defeaters and the relationship between strength and rebuttal. A defeater, in order to rebut a conclusion, must be as strong as the argument supporting the original conclusion. In other words, its premises must be as justified (likely to win an argument) as the ones supporting the conclusion. If a defeater is not as strongly justified as its target, it cannot defeat it but only diminish it. Pollock's theory has been influential in many implemented models of AI reasoning [Chesfievar et al. 2000], but reasoning is not the only use of argument diagramming in AI. One key area is "Computer Supported Collaborative Argumentation" (CSCA), in which the focus is upon developing tools that help people work together using computer infrastructure [Kirschner et al. 2003].

1.3.4.3. Zeno and QuestMap

The diagrammatic reasoning systems used in the public argumentation system Zeno [Gordon and Karacapilidis 1997] are interesting especially because they were intended for actual deliberation, as opposed to education. It was based on the Issue Based Information System (IBIS) framework. Zeno was followed by QuestMap, from "Group Decision Support Systems", an online whiteboard that shows a history of online conversations that led to a decision. QuestMap has been used not only in academic domains, but also for supporting commercial decision making [Conklin 2003, Selvin 2003]. It takes a very broad approach, integrating materials often ignored by more traditional diagramming techniques (including background resources such as articles, spreadsheets, pictures and so on), and allows exploration of a domain in an intuitive and quite unstructured way.

But perhaps the single most successful use of argument diagramming has been with AI tools in education, both in the teaching of critical thinking and argumentation skills themselves, and also as a means to teaching in other subject areas. In the pedagogy of argumentation, there are a number of important examples of tools developed under the auspices of AI (see 1.3.4.6.).

1.3.4.4. Araucaria

Araucaria is a software tool for argument analysis and diagramming developed in 2001 by Chris Reed and Glenn Rowe (see [Reed and Rowe 2004]) based on a representation format, the Argument Markup Language, formulated in XML (see 2.2.2.). The software was developed in Java to achieve interoperability, and is licensed under the GNU General Public License in compliance with the Free Software standards.

The Argument Markup Language (AML) was created to maintain the evolving relationship between text and diagram. AML is designed to be an application of the Argumentation theory in Artificial intelligence and, because it is based on XML (a standard widely used by developers), AML content can be accessed through other software that support XML.

The diagram construction begins by inserting the text of the argument. The next step is to identify each statement that is a premise or a conclusion in the argument and assign it a letter. It is then possible to draw lines representing each inference from the letters representing premises to those representing conclusions. The user may also perform argument reconstruction, supplying missing premises. The resulting diagram has a tree structure, but the software integrates a tool which translates the diagram into either Toulmin diagram or Wigmore diagram, the latter being intended to be used for law cases analysis.

The software is aimed at providing both a pedagogical tool (enhancing the teaching of critical thinking skills thanks to diagramming) and a support for research within the fields of argumentation theory and informal logic. The program was indeed widely used by professionals and scholars in the legal field. Among them are magistrates in Ontario Courts which exploited Araucaria to help with a large volume of relatively simple cases [Reed and Walton 2007, Prakken 2008]. The advantages of using Araucaria in both learning and teaching philosophy was considered by the authors, along with other scholars, in [Rowe et al. 2006].

1.3.4.5. Carneades

Carneades is a formal, mathematical model of argument structure and evaluation, based on the state-of-the-art of argumentation theory in philosophy contained in [Walton 2006], which applies proof standards to determine the acceptability of statements on an issue-by-issue basis [Gordon et al. 2007]. In particular, the Carneades model of argument enables software to inform users whether or not a claim satisfies a proof standard, given the evidence and other arguments which have been put forward by the parties and represented in the diagram. User need not understand the underlying mathematics in order to use Carneades, any more than one needs to understand the formal specification of word processors when writing documents.

Carneades was conceived as a part of the Estrella project. Its background theory and its features will be thoroughly presented in Chapter 4.

1.3.4.6. Other projects

There is clearly a main feature emerging from the AI models of argument diagramming: that of aiding in a variety of educational domains. Belvedere [Paolucci et al. 1995] offered one of the earliest examples, with argument diagrams making concrete the abstract ideas of scientific theories.

Reason!Able [Van Gelder 2001] is designed specifically for pedagogic use, providing a visual framework to organise information, structure reasoning and evaluate evidence. Empirical studies have shown that students who are taught argumentation skills using Reason!Able improve significantly faster and further than those taught using other, traditional techniques [Van Gelder and Rizzo 2001].

More recently, the large SCALE project [Hirsch et al. 2004] has investigated both diagrammatic and dialogic argumentation in high school classrooms. Law pedagogy, in particular, has been a fertile area of investigation.

Vincent Aleven in [Aleven 2003] describes one of the most high-profile systems, CATO, a case-based reasoner that is designed to support law students as they explore cases. It organizes on the basis of issues, and supports a variety of argument structures, but targets text rather than diagrams (for a complete presentation of the software see section 1.4.2.2.).

Diagramming plays a much more central role in systems such as ArguMed [Verheij 2005], where the focus is upon visualizing dialectical argument. For Verheij (the creator of Deflog, see 1.2.7.1.), a range of diagrammatic conventions are required to uniquely represent: support, attack, assumptions, issues, defeat and specificity. One of the key aims of Verheij's work is in capturing Pollock-style undercutters and the subsequent defeat status in his diagrams (shown in the example above by dashed lines and crossed arrows), which makes the approach particularly useful for those AI models derived from Pollock's theory (see 1.3.4.2.).

1.4. Artificial Intelligence and Law

The discipline of "Artificial Intelligence and Law" (AI & Law) joins the state-of-the-art of research in legal subjects and information technologies to provide new instruments for the discipline of law. Under the category of AI & Law are classified all the scientific efforts aimed at creating formal systems that can store conflicting interpretations and consequently propose alternative solutions to a legal issue, as well as systems that use legal precedents to generate arguments by drawing analogies or by distinguishing precedents, or even systems acting as mediator between disputing parties by structuring and recording their arguments and responses. The community has presented very significant research outcomes in this topic since the '80, with different approaches: legal case-based reasoning and more recently also argumentation.

1.4.1. The first computational models of law

Systems to address conflicting interpretations of legal concepts date back to the first years of AI & Law. Thorne McCarty considered a landmark Supreme Court Case in US tax law, concerning differing interpretations of the concept of ownership, and set himself the goal of reproducing in his TAXMAN system both the majority and the dissenting opinions. His intention, however, was not to create a working system, but to gain insight into legal reasoning through a computational model. McCarty's main contribution was the recognition that legal argument involves theory construction as much as reasoning with established knowledge. In [McCarthy 1995] he summarizes his position as follows:

> "The task for a lawyer or a judge in a "hard case" is to construct a theory of the disputed rules that produces the desired legal result, and then to persuade the relevant audience that this theory is preferable to any theories offered by the opponent". [McCarty 1995]

In the same years, another system was developed by Anne Gardner in [Gardner 1987] on the field of offer and acceptance in American contract law. Its task was to *spot issues*: given an input case, it had to determine which legal questions arising in it were easy and which were hard, and to solve the easy ones. The system was rule based, and presented a simpler approach than McCarty's system. One set of rules was derived from a coherent set of 385 law principles contained in the Restatement of Contract Law. The application of these rules was capable of giving a single answer to an issue. A second set of rules was added, taken from case law, common sense and expert opinion. At that point, Gardner distinguished questions by verifying if, linking both the sets of rules to the fact of the case, either a single answer resulted, or no answer, or conflicting answers. In the latter two cases, the question was considered as "hard".

1.4.2. Case-Based Legal Reasoning

Extensive research in AI & Law in the last decades involved the development of a computational model of case-based legal reasoning. The requirements of such a model, set in [Rissland et al. 2003], are:

- A scheme for representing the facts of cases and problems that are legally significant and why;
- A means for assessing the relevance of cases to a problem;
- A mechanism for comparing cases and drawing legal inferences.

These requirements, when met, ensure a big advantage of the model over full-text legal information retrieval system, in that they can draw inferences from the cases and show how they can be used in arguments. At least two representational schemes have been developed, one based on Dimensions [Rissland and Ashley 1987, Ashley 1990] or Factors [Ashley and Aleven 1994, Aleven 1997] which capture stereotypical patterns of fact that tend to strengthen or weaken a side's position on a claim, and another involving Exemplar-Based Explanations [Branting 1991, 1999] which capture an explanation of how the legal conclusions are justified in terms of the facts. See 5.4. for a comparison of those representation schemes with other solutions from AI&Law.

It is worth noticing that, when looking for legal definitions, even statutes are not comprehensive sources. Definitions given in ordinary laws are usually given through examples, and it is necessary to revert to legal doctrine in order to get a deeper explaination of the concept.

1.4.2.1. Dimension-based Representation Schemes: Hypo

Dimensions are highly structured objects, complete with *preconditions* that determine when they apply and *ranges* of possible values that indicate how extreme an example case is in the perspective on that dimension.

The most influential system using dimensions is Hypo, developed by Edwina Rissland and Kevin Ashley (see [Ashley 1990]) in the domain of US Trade Secrets Law. In Hypo, cases are represented according to a number of dimensions. One end of the dimension represents the most favorable position for the plaintiff, while the other end represents the position most favorable to the defendant. In other words, a case’s value on a pro-plaintiff dimension can range from the empty set (the weakest value for plaintiff) to the set of all possible measures (the strongest value for the plaintiff). Typically a case will lie somewhere between the two extremes, thus being favorable to one or the other party, to different extents.

Hypo uses these dimensions to construct three-ply arguments:

- First one party cites a precedent case decided for that side and offers the dimensions it shares with the current case as a reason to decide the current case for that side;
- In the second ply the other party responds either by citing a counter example (a case decided for the other side which shares a different set of dimension with the current case), or distinguishing the precedent by pointing to features which make the current case less favorable to the original side;

- In the third ply, the original party attempts to rebut the arguments of the second ply, by distinguishing the counter examples, or by citing additional precedents to highlight the strengths or hinder the weaknesses in the original argument.

1.4.2.2. Factor-based Representation Schemes: Cato

Years later, Kevin Ashley developed with Vincent Aleven, author of interesting remarks in [Aleven 1997], the Cato system, designed to help law students to learn to reason with precedents. Cato simplifies Hypo in some aspects but extends it others. In Cato, the notion of dimensions is simplified to a notion of factors. A factor can be seen as a specific point of the dimension whose value is binary: it is simply present or absent from a case, rather than present to some degree, and it always favors either the plaintiff or the defendant. Thus if a pro-plaintiff factor applies in a case it represents a strength for plaintiff regardless of the details. In Cato, these factors are organized into a hierarchy of increasingly abstract factors, so that several different factors can be seen as meaning that the same abstract factor is present. The hierarchy allows downplaying (when two factors correspond to the same abstract figure) or emphasizing (when a new figure arises) distinctions.

Cato's model addresses arguments in which two opponents analogize a problem to favorable cases, distinguish unfavorable cases, assess the significance of similarities and differences between cases in light of normative knowledge about the domain, and use that knowledge to organize multi-case arguments. Cato communicates the model to students by presenting dynamically-generated argumentation examples and by reifying argument structure based on the model. It also provides a case database and tools based on the model that help make students' tasks more manageable.

The software was evaluated in the context of an actual legal writing course, finding that instruction with Cato leads to statistically significant improvement in students' basic argumentation skills, comparable to that achieved by an experienced legal writing instructor teaching groups of 4-10 students. However, on a more advanced legal writing assignment, meant to explore the frontier of the Cato instruction, students taught by the legal writing instructor had higher grades, suggesting a need for more integrated practice with the Cato model.

It is interesting that students can learn basic argumentation skills by studying computer-generated examples: it means that an instructional system does not necessarily need to rely on a very sophisticated understanding of students' arguments, which would be a significant obstacle to developing such systems. Also, the model presented novel techniques for using background knowledge to support similarity assessment in case-based reasoning: drawing on its background knowledge, it characterizes and re-characterizes cases in order to argue that two cases are similar or different. This is an important feature in the legal domain which was not previously modeled.

1.4.2.3. Exemplar-Based Explanations and Grebe

Probably the most elaborate representation of cases was produced in the Grebe system in the domain of industrial injury, where cases were represented as semantic networks. It is based on EBEs (Exemplar-Based Explanations, introduced in [Branting 1991]), which represent not only the relevant facts of a case but also aspects of the judge's analysis of their legal significance in justifying her decision. The EBE representation requires the identification of the statutory terms that are disputed in a case, and for each one, the explanation of why the judge decided that the term was (or was not) satisfied in the case. The explanation includes the *criteria*, the facts that the judge deemed legally significant in his determining the statutory norms applicable to the case. These facts are expressed in a relational language and linked to the corresponding statutory term in a semantic network which includes the other statutory terms used by the judge to reach his conclusion.

Grebe uses EBEs to measure the relevance between a problem and a retrieved case. The program matches portions of the network for the new case with parts of the networks of precedents, to identify appropriate analogies. Grebe recursively attempts to apply the statutory rules to the facts, and where particular terms are not further defined by rules, it retrieves cases indexed by those terms and attempts to map the factors from the case onto the problem. The program measures relevance between a problem and a retrieved case as the fraction of the number of unshared and shared factors (here called criterial facts) between them. It selects the best-matching cases and generates a legal argument by analogy (likewise, it is able to distinguish a case from a problem in the light of unshared factors). Grebe had not the task to predict who would win in a problem: it just presents its arguments assigning strength to them depending on the matching of criterial facts. The main limit of the EBE scheme is that it puts a huge effort in representing the cases' and problems' semantic networks, so that the matching of factors would work. This is albeit not easy, according to [Bruninghaus and Ashley 2003], given the vast number of ways to express any such explanation and the difficulty of determining exactly what a judge's rationale is and at what level of abstraction to express it.

1.4.2.4. Issue-Based Prediction of Problem Outcomes

With a legal domain model that relates the factors to issues and a database of case-law represented in terms of (and indexed by) factors, a program called Issue-Based Prediction (IBP) was developed and presented in [Brüninghaus and Ashley 2005]. This software can frame and test hypotheses about which side is likely to win, explain its predictions, and even make the strongest arguments for and against each side.

IBP uses the domain model to identify the relevant issues in a given problem situation, represented as a set of factors. For each issue, it determines if one of the side will win the issue (if all issue-related factors favor the same side) or it poses a hypothesis that the side which has the favor of the majority of retrieved case-law will win. It then tests its hypothesis against the retrieved case-law. If counterexamples are found, IBP determines whether they can be distinguished from the problem situation.

1.4.2.5. Overview on case-law analysis tools

Among all these researches, Hypo in particular was highly influential, being the first to put explicit stress on reasoning with cases while constructing arguments, and providing a dialectical structure in which these arguments could be expressed, anticipating much other work on dialectical procedures. For the purposes of the present research, however, Cato and Grebe are even more important, the first introducing simplifications which enhance the reasoning capabilities of the system and introducing automatic graph construction, the second enriching the data with references to original legal terms.

The main issue related to this kind of representation of the legal contents, is the scarce management of changes. The adaptive nature of case law was explored in [Henderson and Bench-Capon 2001], where the authors considered how understanding of a case law domain would evolve differently depending on the sequence in which cases were presented. Of course, this does have some implications for the use of systems such as Hypo and Cato, which presuppose that case law can be seen as a static body of analysis which can be applied to a new case without adaptation. In Levi’s model [Levi 1948] a period of fluctuation and development is followed by a period of stability, in which the law seems to be well understood and settled. During this period, cases tend to retain a fixed interpretation. Eventually tensions will develop and this will break down, typically through a landmark case. So, Hypo and Cato presuppose that the law be in its period of stability. It is nevertheless necessary to recognize that any analysis will have a lifetime and then need revisiting when the understanding of the domain is changed by some landmark case.

A second consideration by [Henderson and Bench-Capon 2001] concerns the granularity of the analysis. The more abstract the level of analysis, the more likely a new case can be fit into it. On the other hand, results using this coarser classification may be less reliable. The abstract factor hierarchy of Cato helps with this, as far as new aspects can be incorporated as leaves in the abstract factor hierarchy while retaining the structure.

1.4.3. State-of-the-art in AI & Law

In the last years, the research branches of AI & Law detached from the studies on legal decisions and on the *virtual judge* to delve into knowledge representation and automation. Among the most recent work on the subject, [Prakken and Sartor 1996] utilized a very abstract AI framework for representing system of arguments and their relations as developed by [Dung 1995]. In order to be used in AI & Law research, this framework was adapted to legal domain by defining an instantiation of the framework for reasoning with conflicting rules. This inspired several further studies such as [Bench-Capon and Prakken 2010].

[Rissland et al. 2003] constitutes a good summary of the field of Artificial Intelligence and Law, as of 2003. A recent landmark on AI & Law is Giovanni Sartor’s Legal Reasoning: A Cognitive Approach to the Law ([Sartor 2005], which is also the source of the set of distinctions between logics and legal resoning presented in 1.2.8.1.), contained in the *Treaties of Legal Philosophy and General Jurisprudence* of E. Pattaro ([Pattaro et al. 2005]). A major lesson from research on Artificial Intelligence and Law is that legal reasoning cannot be viewed, in general, as the application of some deductive logic, such first-order predicate logic, to some theory of the facts and relevant legal domain.

Regarding the most recent projects in the field, those related to argumentation theory were presented in section 1.3.; others are related to the Semantic Web, and will therefore be presented in section 2.2. after a presentation of the layers of that combination of technologies in section 2.1.

Chapter 2

The Semantic Web

> "The Web is now philosophical engineering. Physics and the Web are both about the relationship between the small and the large."
>
> – Donald Ervin Knuth, *Leaders in computing: changing the digital world*.

> "There are more things in heaven and earth, Horatio, Than are dreamt of in your philosophy."
>
> – William Shakespeare, *Hamlet*, Act 1, scene 5, 166–167.

2.1. Introduction

The revolution brought by Information Technologies allows collecting and processing enormous quantities of data, easily. During the '90s, the Internet introduced a global infrastructure to ensure easy access to these data. However, the huge quantity of information gathered in the World Wide Web, and the possibility ensured for anyone to add more information and make them available to everybody, soon set up an issue on how to manage these data in order to efficiently identify the information useful to the person looking for knowledge on the Internet. As of today, it is possible to navigate only a small part of the web, considering that in order to access a Web page it is necessary to know its URL (Uniform Resource Locator, the *address* of the page) and that only a small part of these pages is reachable through external links, coming from other nodes of the Web. In order to know the URL of the pages that could be of interest to the user, *search engines* were developed since the initial stages of Internet in order to browse through thousands (now millions) of pages in few seconds, looking for keywords indicated by the user and feeding back links to these possibly useful pages. Unfortunately, these engines proved very good at browsing through huge quantity of data, but not as good at evaluating the relevancy of the results. Together with useful pages, matching the purpose of our search, search engines returned "noise", pages whose content has nothing to do with what we´re looking for. That issue didn't depend on the search engines, but was rather a consequence of the Web standards' poor semantic markup.

The standard for web pages is called HTML (HyperText markup Language) and it provides the basic tools to encode the presentation of a text in the web together with a slim structure of metadata (*data about data*, qualifying the content of a part of the text). HTML introduces hypertextual links, providing connection from a document to one other, or from a node of the web to one other. What this essential set of metadata doesn't allow is to enrich these data with information regarding their semantic structure. Without information on its content, machines (such as search engines) cannot manage the data scattered throughout the web pages according to the meaning given to them (which, in turn, depends on the context).

For example, if we are looking for a car on the web, the word "black" that we add together with the car model and other specifications is not *understood* by the search engine as being the color of the car we're looking for, but rather as a mere sequence of letters. In a specialized site, we could be able to choose "black" among a list of possible colors so that the machine knows that we are not looking just for a sequence of letters, but rather for any piece of text whose meaning is "black colour of the car". These information, however, are not encoded in the basic Web technologies.

Since 1999 the World Wide Web Consortium (W3C), founded and directed by Tim Berners-Lee with the goal of promoting standards that ensure interoperability in the Web (see [Berners-Lee et al. 2006]), focused its research on resolving this issue and gave birth to an approach called "the Semantic Web". The purpose of that approach is to develop an extension of the Web that will transform it in a net of machine-readable data. The goal is to render computers capable of inferencing new knowledge on the basis of given information. Under a technical point of view, this is made possible by managing the structure of information so that documents do not remain islands of meaningless data, rather becoming "open databases" capable of enriching the information available to a software.

2.1.1. Linked Data

The term *Linked Data* defines a set of best practices for exposing, sharing, and connecting pieces of data, information, and knowledge on the Semantic Web. It originates from the DBpedia initative: a crowd-sourced community effort introduced in [Bizer, et al. 2009], whose aim is to extract structured information from the Wikipedia articles in order to make it available on the Web for the development of new mechanisms of navigating, linking and editing the Wikis.

Linked Data exploits URIs and RDFa (see 2.2.1. and 2.2.3.)to enhance the connections between documents available on the Internet with metadata describing their content, thus improving the availability of these data. The ideal audience of such a web of data, besides the human web surfer looking for specific information on the Internet, is the web application of civic interest which, thanks to this improvement, could rely on a solid knowledge base which could be automatically retrieved and managed.

An example (in the field of medicine) of the utility of such an advancement was given by Tim Berners-Lee (creator of the World Wide Web) in [Berners-Lee 2009] while launching the Linked Data initative. A lot of data is available on Alzheimer's drug discovery, because scientists in that field consider it a good way of getting information out of the "isolated silos" of their research center's computers. Those scientists had genomic data in one database in one building, and protein data in another. Having linked those data it is possible to submit queies to the computer such as: “What proteins are involved in signal transduction and also are related to pyramidal neurons?”. Submitting this query to Google would return 223,000 results, but none of them significant, because nobody has asked that question before. Querying the Linked Data would return 32 results, each of which is a protein which has the requested properties, and can be looked at. As [Berners-Lee 2009] puts it,

> “The power of being able to ask those questions of a scientist, those questions which actually bridge across different disciplines is really a complete change. It's very important. The power of the data that other scientists have collected is locked up and we need to get it unlocked so we tackle those huge problems”. [Berners-Lee 2009]

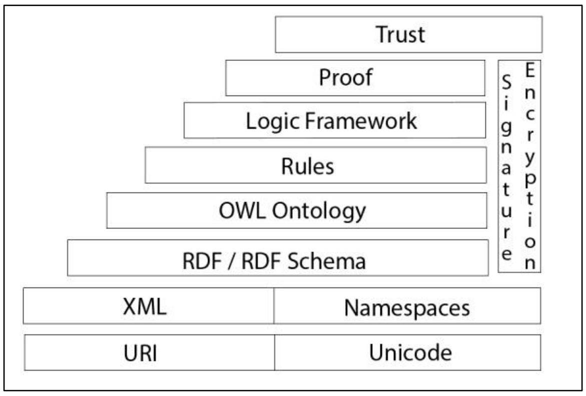


Fig. 2.1 – The Semantic Web stack of technologies.

In the legal field, such an improvement could prove to be fundamental in enhancing the management of legal documents. If legal documents were self-explaining and inter-operable, their content being semantically linked to contents of official law repositories of the States available online, the real concept of “legal procedure” could change. Legally relevant activities performed (or represented) on the web could be linked as raw data, being watchable and modifiable by other tools representing other legal activities. However, in order for this to happen, the issue of the knowledge acquisition bottleneck has to be resolved (see 5.2.1.).

2.2. The Layers of the Semantic Web

The evolution planned by the Semantic Web goes from the concept of “information retrieval” (producing documents which are relevant to a query) to that of “data retrieval” (giving the correct answer to a question). It has been represented in 2001 by Tim Berners-Lee as a 7-layer cake made of 9 elements (Figure 2.1), where every layer is connected to the others through three typologies of information. The architecture has to be presented from the bottom of the cake, because the upper layers are built on standards and elements formalized in the lower ones. The levels of the Semantic Web pyramid are thus the following:

- **Identifiers and character set**, providing a “name” for entities on the web and a set of character to spell those names and communicate information about their content;
- **Syntax**, setting the pattern for the additional description of the entities on the web;
- **Data interchange**, allowing to establish relations among different sets of entities on the web;
- **Ontologies**, introducing restrictions to these relations in order for the knowledge base to be valid (free from errors) and consistent (free from contradictions).

- **Unifying logic**, performing complex reasoning on the knowledge base which involves different sets of entities on the web;

- **Proof and trust**, ensuring that the knowledge base and the reasoning performed on it are authored, substantially correct, and up-to-date.

It is also to be noticed that the top two layers (*proof* and t*rust*) require technological innovations and standards that do not exist yet. For this reasons, the “label” of these layers describes not the instrument, but the socio-cultural impact that the transformations brought by these two technologies will bring to the web.

In the following sections the layers of the Semantic Web will be presented, describing their functions and introducing the standards which were reached in the last years, as well as interesting implementation of these technologies in the legal field.

2.2.1. Unicode and URI

These two standards represent the “basic bricks” for the construction of standard semantically-enriched documents.

Unicode is a standard for the consistent encoding, representation and handling of text expressed in most of the world’s writing systems. The origins of this language date back to 1987, when Joe Becker, Lee Collins and Mark Davis from Apple started investigating the practicalities of creating a universal character set[6]. In August 1988, Joe Becker published a draft proposal for an *international/multilingual text character encoding system, tentatively called Unicode*. Actually, Unicode consists of a collection of more than 109.000 characters together with character properties (i.e. upper/lowercase).

The URI (Uniform Resource Identifier) is a string of characters used to identify a name or a resource on the Internet. Such identification allows interaction with other representation of that resource over a network. URIs shares its origins with URLs: in 1994, Tim Berners-Lee’s proposals for *HyperText* introduced the idea of a short string representing a resource that is the target of a hyperlink. At the time, people referred to it as a 'hypertext name' or 'document name' [Palmer 2009] . URIs, in fact, represent a conjunction of URN (Uniform Resource Name, giving a unique name to a resource) and URL (Uniform Resource Locator, giving a unique *address* to the resource). By combining information on the *identity* of a resource and on its *location* allows for other document to refer to it in an unambiguous way. Once every relevant resource on a document is provided with its URI, the data contained in the document can be used together with similar data coming from other document to make inferences and provide correct answers for data retrieval purposes. Thanks to URIs it is possible to refer not just to a document as a whole, but also to parts of them, or to their content.

2.2.2. XML

---

[6] See History of Unicode, http://www.unicode.org/history/index.html.

XML (eXtensible Markup Language) is a markup language whose origins are closely related with HTML. It is a descendant of SGML (see [Goldfarb 1991]), the Standard Generalized Markup Language, invented by Charles F. Goldfarb, Ed Mosher, and Ray Lorie at IBM in the 1970s and developed by several hundred people around the world until its adoption in 1986 as ISO standard 8879. SGML is a powerful semantic and structural markup language for text documents first applied in the U.S. military and government, in the aerospace sector, and in other domains that needed ways of efficiently managing technical documents that were several thousands of pages long.

HTML was the most famous application of SGML but did not offer anywhere near the full power of the original language, since it restricts authors to a finite set of tags designed to describe web pages, in a presentational-oriented way. HTML would be the best choice to do web pages, but it would never be used, for example, to exchange data between incompatible databases. For these tasks, SGML was the obvious choice but there was a problem: it is a very complicated language, covering many special cases. It is so complex that almost no software implemented it fully, and programs rather relied on differents subsets of SGML which were often incompatible to each other. In order to address these issues, in 1998 XML 1.0 was issued (W3C Recommendation of February 10, see [Goldfarb 1998]). Built as a lite version of SGML retaining most of SGML's power while removing a lot of the features that had proven redundant, too complicated to implement, confusing to end users or simply not useful over the previous 20 years of experience, XML was an immediate success and is used since then in a wide variety of domains. Successive additions were namespaces in XML, an effort to allow markup from different XML applications to be used in the same document without conflicting, and the Extensible Stylesheet Language (XSL), an XML application for transforming XML documents into a form that could be viewed in web browsers.

Being the most common tool for data transmissions between all sorts of applications, nowadays XML is as important for the Web as HTML was to its foundation. Applications in the legal domain relevant for the present research are introduced in section 2.4.1. For a complete coverage of XML-related projects in the legal field see [Sartor et al. 2011].

2.2.2.1. Main features of XML

An XML document contains text – never binary data – and can be opened with any text editor. An XML document can be a very short or a very long document, but it has to be a well-formed (see below 2.2.2.2.) XML document in order to be correctly read and “understood” by an XML parser. The shortest possible example of XML document is the following:

```
<person> Freddie Mercury </person>
```

This document could be contained in a .xml file, and it wouldn´t matter which is the name of the file as far as the parser is concerned. It could even not be in a file at all (for example it could be a record in a database) or it could be generated on the fly in response to a browser query.

There are three important components in the example seen above: `<person> </person>` are *tags* (start/end-tag), while everything between those tags is called *content*. Altogether, this is an *element* named person. These tags are different from those of HTML because XML allows making up new tags. It is therefore possible to furtherly define the information in total freedom, for example as follows:

```
<person><first_name>Freddie</first_name><surname>Mercury<surname
> </person>
```

This example shows another property of XML: it allows creating a hierarchy of terms, where i.e. `<person>` is a parent while `<first_name>` and `<surname>` are child elements. One thing that XML doesn't allow is overlapping tags: this means that an element beginning inside another element must also finish inside that element. This allows the building of semantic trees descending from a root (or parent tag) and furtherly specifying bits of information. This is very useful for writing database entries, but XML is also very powerful in free-form, narrative document tagging:

```
<person><name><first_name>Freddie</first_name><surname>Mercury</
surname> </name> was a <nationality>British</nationality>
<profession>musician </profession>,
<profession>singer</profession> and <profession>songwriter
</profession>, best known as the <role>lead vocalist</role> of
the rock band <band>Queen</band>.
```

XML elements can also have attributes. An attribute is a name-value pair attached to the element's start-tag. For example, we can add information on the person we are talking about:

```
<person born=”1946-11-05” died=”1991-11-24”> Freddie Mercury
</person>
```

When building databases the choice of which information is to be represented as elements, and which through attributes, can be subject to debate. Instead, when marking up free-form texts, the distinction is clearer: the information already present in the text should be kept and highlighted with the proper element, while all information which is implicit in the text (i.e. the language of the text) should be added through attributes.

2.2.2.2. Well-Formedness

Every XML document, without exception, must be well-formed. This means it must adhere to a number of rules, including (but not limited to) the following:

– Every start-tag must have a matching end-tag;

– Elements may nest, but may not overlap;

– There must be exactly one root element;

– Attribute values must be quoted;

– An element may not have two attributes with the same name;

– Comments and processing instructions may not appear inside tags;

– No unescaped < or & signs may occur in the character data of an element or attribute.

In addition, XML-Schemas and Document Type Definitions (DTDs) provide a more specific set of rules concerning the elements which is expected to be found in a well-formed document of that type. Validation is the act of checking the correctness of an XML document according to pre-defined structural rules expressed in one or more DTDs and XML Schemas. The validation verifies whether the XML document contains, in number and position, all the expected elements of the type this document is an instance of. If the parser detects an error, he reports it but it is not allowed to try to fix the document on itself, even if it involves only trivial operations. This way of checking the correct construction of the XML documents is the key to its success: it allows checking the integrity of the document without introducing a machine intervention in the XML files that could bring to inefficacies (for example if there is a bug in the parser). At the same time, the well-formedness sets the basis for the consistency check of the upper layers in

Jack -> son_of -> Maria

Farrokh Bulsara ->birthname_of-> Freddie Mercury

Fig. 2.2 – Basic RDF triple

the Semantic Web stack of technologies.

2.2.3. RDF and RDF Schema

The elements introduced through XML tagging are metadata (data on data): they describe a resource in such a way to make it more comprehensible to the user. Normally, metadata are descriptive notations adding information on the document (author, title, abstract, file type, copyrights, version number…). These data are useful for managing resources, to archive and identify information. However, the most important function of metadata is to promote interoperability, by maintaining a combination of heterogeneous resources on different platforms without losing relevant information.

In fact, the concept of *open data* refers to an important characteristic of the Semantic Web, namely the disposability of data and the consequent possibility to identify and use these data (see [Decker et al. 2000] and 5.2.). The Semantic Web is therefore an extension of the Web where the concept of “linking” is much more developed in order to work with a relational model of data, where the single “link” is not a blind bond between two documents, but rather is a conceptual relation between two individuals. In fact, on the Semantic Web every “piece” of information is identified through an URI.

The URIs are put in relation to each other through a language called RDF (Resource Description Framework, see [Nejdl et al. 2000]), which allows to build assertions through triples formed by a subject, a predicate and an object. RDF was conceived to represent information about resources in the World Wide Web, mainly metadata about Web resources (such as title, author, modification date of a Web page). However, by generalizing the concept of a "Web resource", RDF can also be used to represent information about things that can be identified on the Web, even when they represent something that is outside of the Web. Examples include information about items available from online shops (information on their characteristics, prices and availability) and the description of a Web user's preferences for a service.

RDF is intended for situations in which this information needs to be processed by applications, rather than being only displayed to people. RDF provides a common framework for expressing this information so it can be exchanged between applications without loss of meaning. This means that the information may be made available to applications different from those for which it was originally created. Since it is now a standard framework, application designers can take advantage from the availability of common RDF parsers and processing tools.

RDF is based on the idea of identifying things using Web identifiers (URIs), and describing resources in terms of simple properties and property values. The elements are put in relation to each other through triples formed by a subject, a predicate and an object represented in Figure 2.2. This enables RDF to represent simple statements about resources as a graph of nodes and arcs representing the resources (through their URIs) and their properties and values. RDF also provides an XML-based syntax (called RDF/XML) for recording and exchanging these graphs.

The combination of RDF and XML not only grants the enrichment of a document with information processable by a machine, but also allows to build a conceptual graph, connecting the information together and allowing to perform automatically operations more complex than a simple search on the document's content.

2.2.3.1. RDFa

RDFa is an extension of the basic RDF language, providing a set of attributes that can be used to carry metadata in an XML language (hence the *a* in RDFa). These attributes are:

- **About**: a URI or CURIE specifying the resource the metadata is about;
- **Rel** and **rev**: specifying a relationship and reverse-relationship with another resource, respectively;
- **Src**, **href** and **resource**: specifying the partner resource;

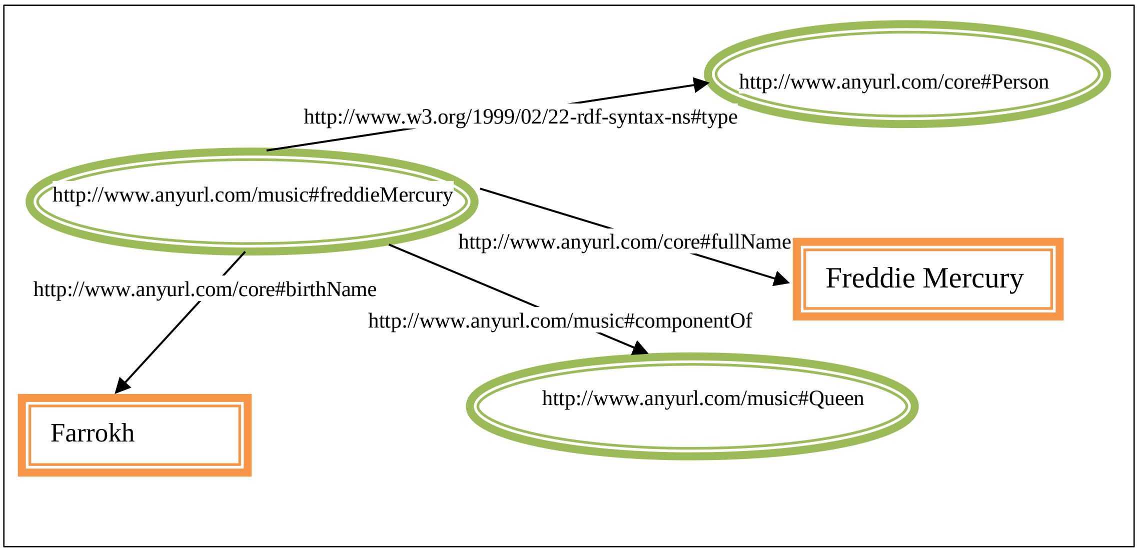


Fig. 2.3 – A small RDF graph.

- **Property**: specifying a property for the content of an element or the partner resource;
- **Content**: optional attribute that overrides the content of the element when using the property attribute;
- **Datatype**: optional attribute that specifies the datatype of text specified for use with the property attribute;
- **Typeof**: optional attribute that specifies the RDF type(s) of the subject or the partner resource (the resource that the metadata is about).

RDFa reportedly meets a set of *principles of interoperable metadata* met by RDFa[7]:

- **Publisher independence**: each site can use its own standards;
- **Data reuse**: data are not duplicated, separate XML and HTML sections are not required for the same content;
- **Self-containment**: the HTML and the RDF are separated;
- **Schema modularity**: the attributes are reusable;
- **Evolvability**: additional fields can be added and XML transforms can extract the semantics of the data from an XHTML file.

2.2.3.2. Dublin Core

The Dublin Core set of metadata elements is an ISO Standard providing a fundamental group of text elements which can be used to describe and catalogue multimedia resources such as books, video, sound, image or text files, and Web pages. Being it a standard on both the fields of library science and computer science, the metadata records based on Dublin Core are also used in cross-domain information description. The work originated during the 1995 invitational OCLC/NCSA Metadata Workshop, hosted by the Online Computer Library Center (OCLC) and the National center for Supercomputing Application (NCSA). The expansion and development of this standard is brought forward by the Dublin Core Metadata Initiative (DCMI, see [Weibel 2009]), which promotes widespread acceptance of metadata standards and practices through working groups, global conferences and workshop, standards liaison and educational efforts.

Being it a "core" work, its elements are broad and generic, usable for describing a wide range of resources. The Simple Dublin Core Metadata Element Set (DCMES) consists of 15 metadata elements:

- **Title**: a name given to the resource;
- **Creator**: an entity primarily responsible for making the resource;

[7] See RDF Primer http://www.w3.org/TR/xhtml-rdfa-primer/).

– **Subject**: the topic of the resource;

– **Description**: an account of the resource;

– **Publisher**: an entity responsible for making the resource available;

– **Contributor**: an entity responsible for making contributions to the resource;

– **Date**: a point or period of time associated with an event in the lifecycle of the resource;

– **Type**: the nature or genre of the resource;

– **Format**: the file format, physical medium, or dimensions of the resource;

– **Identifier**: an unambiguous reference to the resource within a given context;

– **Source**: a related resource from which the described resource is derived;

– **Language**: a language of the resource;

– **Relationn**: a related resource;

– **Coverage**: the spatial or temporal topic of the resource, the spatial applicability of the resource, or the jurisdiction under which the resource is relevant;

– **Rights**: information about rights held in and over the resource.

The standardization process of these elements started in 1998. Notions of best practice in the Semantic Web evolved to include the assignment of formal domains and ranges in addition to definitions in natural language. Domains and ranges specify what kind of described resources and value resources are associated with a given property, expressing meanings which are implicit in natural-language definitions in an explicit form that is usable for the automatic processing of logical inferences. When a given property is encountered, an inferencing tool may use information about the domains and ranges assigned to a property in order to make inferences about the resources described thereby.

Since January 2008 Dublin Core includes formal domains and ranges in the definitions of its properties (see [Weibel 2009]). In order not to jeopardize backward-compatibility with existing implementations of "simple Dublin Core" in RDF, domains and ranges have not been specified for the fifteen properties of the `dc:namespace`. Rather, fifteen new properties with "names" identical to those of the Dublin Core Metadata Element Set Version 1.1 have been created in the `dcterms:namespace`. It is thus possible to choose between the legacy `dc:` variant and the `dcterms:` variant, depending on application requirements. Over time, however, implementers were encouraged to use the semantically more precise `dcterms:` properties, as they more fully follow emerging notions of best practice for machine-processable metadata.

2.2.4. Computational Ontologies

RDF/XML constitutes the basis for exchange of information between different application with no loss of meaning, representing in a machine-readable way not only the semantics of pieces of text (through URIs) but also the links between different resources. In this way, applications are able to represent these resources in a graph which constitutes a first step towards a formal representation of the knowledge contained in the document. However, in order to make a knowledge field explicit, a further step is necessary, namely the linking of the concepts to logic rules concerning their usage. This task is accomplished by ontologies.

In philosophy, an ontology is a theory on the “essence”, or “nature of existence” of different kinds of (material and immaterial) objects. In technical theory, an ontology is an agreement based on a common vocabulary (taxonomy) representing the conceptual basis upon which different subjects operate. The term has become common between the artificial intelligence and web research communities since [Gruber, 1993] to indicate a document formalizing the relations between terms. “Computational ontology” is used in the community with two meanings: a collection of data and vocabularies (a Tbox defining terms and relations between them and an Abox providing assertions on these terms) or a collection of only vocabularies (Tbox). For a complete presentation of the concept of Computational Ontologies see [Casellas 2011].

An ontology is present also in Figure 2.3, shown in the last section: the *is_a* relation is in fact one of the basic elements constituting an ontology. Relations such as *is_a* are built-in logic connectors of language. The typical Web ontology contains a taxonomy and a set of inference rules. The taxonomy defines objects and relations between them: in this way, the meaning of the terms contained in a document can be defined through pointers linking to an ontology. A semantic agent looking for mercury and finding Freddie Mercury must be able to understand that we are not looking for a person, but rather for a chemical element or a planet.

The ontology is a semantic tree introducing restrictions on terms in the form of relations. Through the use of description logic it is possible to express any object or concept in a formal way. Ontologies contain the specifications of those objects or concepts which are necessary for the understanding of the knowledge domain, its vocabulary, the way concepts and vocabularies are connected and the way classes, instances and their properties are defined.

An ontology can be formal or informal. Only formal ontologies are machine-readable (which means that a computer can also make inference from them) but it is quite hard to apply those formal construct to practical knowledge. Nevertheless, a first important issue that can be solved through ontologies is that of synonyms: two different databases may well use different elements to represent the same concept, and in this case it is necessary for an application combining or comparing information in these two databases to know that more terms can refer to the same concept. Ontologies can build relations between data using a common language, thus enhancing sharing of information between different languages.

2.2.4.1. OWL

Ontologies are written in either RDF Schema, OWL (Ontology Web Language[8]) or other specific languages such as DAML, DAMOIL. The most implemented ontology is actually Wordnet, describing concepts (synonyms, opposites, relations between concepts) in several

---

[8] See OWL Web Ontology Language http://www.w3.org/TR/owl-ref/.

languages. It is capable of doing “conceptual dependency searches”, such as mereology (part-whole) or metonymic (content-container) relations.

It is possible for anyone to create a reference ontology, hoping it to be shared by others, or to choose an existing one, thus embracing an other’s perspective. Usually, ontologies are created within a scientific community. An issue raised mostly by philosophers and sociologists is about the fact that every person defines the world his own way, and that creating these definitions is an important part of the job of any researcher in human or social sciences. There is therefore a high risk of creating overlapping or conflicting conceptualizations, while on the other hand a single, universal ontology – which is the goal to be reached under a certain perspective – could bring to an excessive rigidity in interpretation of facts and phenomena. An enumeration of pros/cons in legal domain is contained in [Mommers, 2010].

This trade-off between conflicts and rigidity can nevertheless become an advantage. In fact, it leads to a multiplicity of ontologies, because a variety of classifying methods can achieve a better correspondence between terms and concepts than a unique vocabulary: moreover, it is possible to create meta-ontologies specifying correspondence between synonyms, or establishing relations between different vocabularies. Often, different research groups develop similar concepts independently and it is therefore very useful to describe the relation between these concepts. Facing several different ontologies is a normal scenario for any application managing semantic contents, and it will be a task of the software itself to highlight differences, thus helping the user in understanding the concept in question.

Another element fostering plurality, which in turn distinguishes Semantic Web theorists form Artificial Intelligence researchers, comes from the consequences of the so called Open World Assumption, by which the truth of an assertion is independent from the knowledge base. In other words, this means that if an assertion is not known to be true it cannot infer that it is false (As it would happen following the negation-as-failure axiom of the Closed World Assumption, see 5.2.2.5.).

2.2.4.2. The main elements of OWL

The building blocks of an OWL ontology are the following:

Classes provide an abstraction mechanism for grouping resources with similar characteristics. Like RDF classes, every OWL class is associated with a set of individuals, called the class extension. In addition, the following types of class description add semantic information on the class:

- The **class identifier**, the URI reference for the class;
- A list of **equivalent** element, enumerating the (combination of) classes or properties which have the same class description. The equivalent elements may be expressed by an axiom (i.e. elements that have a certain propert with a certain range);
- **Union of**, indicating that the class description is the sum of two or more class descriptions;

- **Intersection of**, indicating that the class description is equivalent to the intersection of two or more class descriptions;
- **Complement of**, indicating that the class includes all the individuals that are not member of one (or more) class descriptions;
- **Subclass of**, indicating that all the individuals of this class are also member of one (or more) class descriptions;
- **Disjoint with**, asserting that the class extensions of the two class descriptions involved have no individuals in common.

Instances are the individuals identified in the class extension and in the class description, defined by elements such as:

- **Same as**, identifying the individuals which have the same identity (two URI references that actually refer to the same thing);
- **Different from**, identifying two URI references that refer to different individuals.

Properties are divided into two categories: *datatype properties* link individual to individuals, while *object properties* link individuals to values. Its main attributes are:

- **Domain**, asserting that the subjects of such property statements must belong to the class extension of the indicated class description;
- **Range**, asserting that the values of this property must belong to the class extension of the class description (for *object properties*) or to data values in the specified data range (for *datatype properties*);
- **Inverse of**, asserting that the property is the inverse of another property: its *range* corresponds to the *domain* of the target property, and its *domain* corresponds to the *range* of the target property;
- **Subproperty of**, similar to *subclass of*, asserts that the individuals which constitute the subject or the value of the property are also subjects (or values) of another property.

Annotations add information (such as label, description) to the element of the ontology. In this way, it is possible to make clear the modelling intentions of the ontology creator, explaining the purpose and intended meaning of the OWL code using natural language.

The OWL language includes many more elements, representing the various aspects of the logics supported (AllValuesFrom, SomeValuesFrom introducing elements of description logics in class descriptions; transitive and functional attributes for properties, and so on).

2.2.4.3. OWL 2.0

OWL2 is a new version of the Ontology Web Language, released in 2009. OWL2 has a very similar overall structure to OWL. Looking at Figure 2.4, almost all the building blocks of OWL2 were present in OWL, albeit possibly under different names. The central role of RDF/XML and the role of other syntaxes has not changed, and backwards compatibility with OWL is complete, with identical inferences in all practical cases.

OWL 2 adds new functionalities, including keys, property chains, richer datatypes and data ranges, qualified cardinality restrictions, asymmetric, reflexive, and disjoint properties, and enhanced annotation capabilities (the most relevant of whom will be presented in Chapter 3). It also defines new profiles (in addition to the existing OWL DL, OWL Full and OWL Lite the new OWL EL, OWL 2 QL and OWL 2 RL dialects were added) and a new syntax. Figure 2.5 shows the complete table of available syntaxes for OWL.

Name of Syntax	Specification	Status	Purpose
RDF/XML	Mapping to RDF Graphs, RDF/XML	Mandatory	Interchange (can be written and read by all conformant OWL 2 software)
OWL/XML	XML Serialization	Optional	Easier to process using XML tools
Functional Syntax	Structural Specification	Optional	Easier to see the formal structure of ontologies

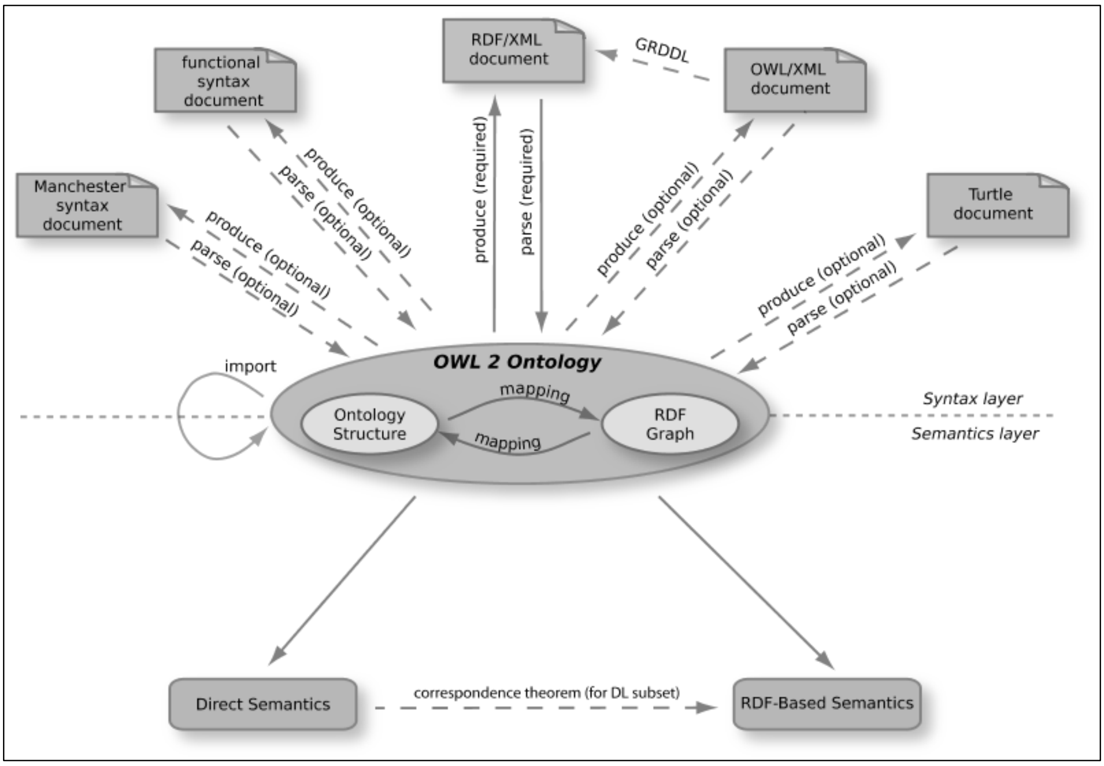


Fig. 2.4 – Structure of OWL 2

Manchester Syntax	Manchester Syntax	Optional	Easier to read/write DL Ontologies
Turtle	Mapping to RDF Graphs, Turtle	Optional, Not from OWL-WG	Easier to read/write RDF triples

Fig. 2.5 – Syntaxes of OWL 2

2.2.5. Rules and Logics

The term *rule* has different meanings in different fields. In Kelsen's definition of Law [Kelsen 1945] it is defined as as "an order of human behavior" The author explained the word *order* to mean *a system of rules*, and his contribution to the legal positivism represents a fundamental step towards the modern vision of normative rules; in fact, while agreeing with the positivists' conception of law as *posits* (social constructions) rather than derivations from reasons, he argues that law always needs a normative base: a *Grundnorm* (basic norm), from which the legitimacy or validity of all the laws derive.

In ITs, rules are of the form of an implication between an antecedent (body) and consequent (head). The intended meaning can be read as: whenever the conditions specified in the antecedent hold, then the conditions specified in the consequent must also hold.

By combining properties expressed in RDF and axioms expressed in OWL (or other ontology language) it is already possible to infer new knowledge from the existing information. In Tim Berners-Lee Semantic Web Cake, this is the logic layer, where "reasoning" takes place. Reasoning means inferring new knowledge: in order to achieve that, specific software must be able to connect the terms. Common reasoners include Pellet, Fact++, Hermit. Reasoning can be efficacious only if the software is able to recognize synonyms, translating the properties of a term into its analogous, and then checking the known data to identify useful information.

The Semantic Web Rule Language (SWRL) is a proposed language for the Semantic Web that can be used to express rules as well as logic, combining OWL DL or OWL Lite with a subset of the Rule Markup Language (RuleML). The logic layer, however, is the first layer of the Semantic Web stack which has not reached a shared standard yet: the debate is still open from here to the top of the pyramid.

2.2.5.1. Closed/Open World Assumption

We introduce here an important distinction, central to the present research: the distinction between the closed world assumption and the open world assumption.

The *closed world assumption*, formalized in [Reiter, 1978], is the presumption that what is not currently known to be true is false. An important concept inside the closed world assumption is the concept of *negation as failure*, which implies believing false every predicate that cannot be proved to be true. In the knowledge management arena, the closed world assumption is used in at least two situations: when the knowledge base is known to be complete (e.g., a corporate database containing records for every employee), and when the knowledge base is known to be incomplete but a "best" definite answer must be derived from incomplete information.

The opposite of the closed world assumption is the *open world assumption*, stating that lack of knowledge does not imply falsity. The most important implication of this assumption is that, if a rule has a negative requirement (i.e. the body of the rule contains *(not) knowlable*) and the Knowledge Base contains no information on the element (i.e. *knowable*), the rule will not trigger. In order for the rule to apply, the Knowledge Base must explicitly contain the negative statement (i.e. *(not) knowable*).

This distinction determines the understanding of the actual semantics of a conceptual expression with the same notations of concepts. A successful formalization of natural language semantics cannot avoid an explicit revelation of the logical backgrounds based on either closed or open world assumption. See 5.1. for a more complete discussion on the topic in the legal field.

2.2.6. Proof and Trust

The top two layers of the Semantic Web layer cake are still relatively unexplored, and no standard on these two subjects has arisen yet.

Goal of the proof layer is to give a proof that an answer found in the Semantic Web is correct under different perspectives, such as how it was derived (logic), on which data (sources) and by who (trust). As will be seen in the next chapters, standardization in document modelling, markup, and knowledge representation are explicitly aimed at covering part of that issue: with a correct implementation of the underlying layers it is in fact possible to render explicit the sources and logic paths involved in an answer of the Semantic Web.

To fully implement trust is a whole different issue: it involves problems of authentication, interoperability, scalability, security, exposure control and so on. Some literature on those themes can be found in [Wahlforss and Ljung 2008] and [Artz and Gil 2007].

2.3. The four layers of a legal document

Ever since its creation, the World Wide Web was conceived as a web of documents. The URLs for HTTP (HyperText Transfer Protocol) were identifiers for web pages consisting essentially in text and images. Additional applications such as Flash allow adding sounds or videos inside the URL, but those documents were considered only as a whole. The Web 2.0 brought a revolution into that concept, and RDFa introduced the possibility to use URLs to create a web made of links not just between documents, but also between the content of those documents: the data (this new approach to knowledge construction in the web has been analysed in 2.1.1.).

The present section focuses the attention on the original web of documents. In fact, representing a set of legal documents (such as case-law) in a net-like infrastructure is not an effortless task. In order to represent a decision where some contracts and laws are concerned, it is necessary to represent those documents differentiating between different versions (i.e., two versions of a law, before and after an amendment). In order for the knowledge base to be handled effectively, there can be no mistake between the abstract conception of a contract clause, a specific clause of a specific contract, and a clause contained only in one copy of that contract.

This issue does not concern the content of the documents involved in a judicial decision, but rather concepts such as succession over time and serialization. It does not concern, for example, the status of a law as the enactment of some general guidelines contained in a prior law or on a

European decision. It only concerns the management of the production and distribution processes that every externalization of concepts (every document) undergoes.

In order to introduce these distinctions into the representation of the legal documents, the standard on the field of general literature will be analyzed: the classification of the Functional Requirements for Bibliographic Records. In 1992 the IFLA (International Federation of Library Associations and Institution) started the study group on Functional Requirements for Bibliographic Records (FRBR) with the purpose of developing an entity relationship model as a generalized view of the bibliographic universe, intended to be independent of any cataloguing code or implementation. The standard proposal, available in (International Federation of Library Associations and Institutions, 2013) includes a description of the conceptual model (the entities, relationships and attributes), a national level bibliographic record for all types of materials, and user tasks associated with the bibliographic resources described in catalogues, bibliographies, and other bibliographic tools. The most interesting part of the conceptual model of FRBR is the identification of the four layers of documents, or, in other words, of the four possible “targets” that can be intended when a document refers to another document: it can in fact be possible that, by pointing at some other document, a source document intends to refer to the original or general idea behind it (a *work*), to that version of the idea (an *expression*), to the series of documents the target is an exemplar of (a *manifestation*) or to that single exemplar (an *item*). Which of these layers is the intended target of the reference, however, is not always easy to determine. A clear distinction of the borders between those layers, especially in the legal field, is a necessary starting point. The following paragraphs are therefore aimed at defining the shapes that these classes assume in the legal field (see also [Cuninghame 2009]).

2.3.1. FRBR Classification: Work

The first entity of the classification is work: a distinct intellectual (or artistic) creation. A work is an abstract entity: there is no single material object one can point to as the work. When speaking of Homer's Iliad as a work, the point of reference is not a particular recitation or text of the work, but the intellectual creation that lies behind all the various expressions of the work. Because the notion of a work is abstract, it is difficult to define precise boundaries for the entity. The concept of what constitutes a work and where the boundary between one work and another lies may in fact be viewed differently from one culture to another. Consequently the bibliographic conventions established by various cultures or national groups may differ in terms of the criteria they use for determining the boundaries between one work and another. Variant texts incorporating revisions or updates to an earlier text, abridgements or enlargements of an existing text, and translations from one language to another are usually viewed as expressions of the same work. Here is a pair of examples:

w1 Ridley Scott’s “Blade Runner” motion picture

e1 the original version

e2 the director’s cut

w2 Philip K. Dick’s “Do Android Dream of Electric Sheeps?”

e1 the original version in English language e2 the Italian version in Italian language

In the legal field, modifications of single clauses (or articles) of an existing law create a new expression of the same work:

w1 The Italian Civil Code

e1 The original version of the royal decree n. 262 of 1942

e2 The version modified by law 151 of 1975

e3 The version modified by decision 245/2011 of the Constitutional Court

By contrast, there must be a certain threshold when the degree of independent intellectual or artistic effort involved in the modification of a work where the result is viewed as a new work. Thus paraphrases, rewritings, adaptations for children, parodies, musical variations on a theme are considered to represent new works. Similarly, adaptations of a work from one literary or art form to another (e.g., dramatizations, adaptations from one medium of the graphic arts to another, etc.) are considered to represent new works. Abstracts, digests and summaries are also considered to represent new works. In the previous example:

w1 Philip K. Dick's "Do Android Dream of Electric Sheeps?"

w2 Ridley Scott's "Blade Runner" motion picture

In the legal field, paraphrases of legal norms made by doctrine or jurisdiction are viewed as new works. Also sentences summarized in case-law collections constitute new works.

w1 Art. 1341 of Civil Code

w2 The chapter "The discipline of art. 1341 of Civil Code" by A. Torrente and P. Schlesinger

…

w1 Decision of Cassation Court n. 1348 of 1998

w2 The summary in the paragraph "1348/1998" of the Cassation Court case-law collection

Following some critics on the original FRBR classification presented above, among which are [Weihs 1998] in the need for the entity "superwork" and [Smiraglia 2002] on the *instantiation network*, a new version of the classification, called $FRBR_{00}$, enriched the Work layer creating the new classes *Complex Work*, *Individual Work*, *Self-Contained Expression*, *Container Work*, *Aggregation Work*, *Serial Work*, *Publication Work*, *Performance Work*, *Recording Work*.

On a pragmatic level, defining work as an entity in the model serves a number of purposes. It enables to give a name and draw relationships to the abstract intellectual or artistic creation that encompasses all the individual expressions of that work. Thus, when we describe a work of legal doctrine dealing with Civil Code's discipline of consumer contracts, for example, we are able to relate the work of doctrine to the work that it treats as its subject. By naming Civil Code and defining the relationship between it and the work of doctrine, we are able to indicate that the subject of the work of doctrine is in fact the abstraction we know as the Civil Code and not any specific expression of that work.

Defining work as an entity also enables to establish indirect relationships between expressions of the same work in cases where we are unable to draw direct relationships between individual expressions. For example, there may exist many amendments of a work (e.g., discipline on Consumer Contracts), and it the text that has served as the basis for a given amendment may not always be directly specified (the amendment may in fact e only an indirect result of the application of the new norm). In that case we do not draw a direct relationship between individual expressions of the work (i.e., between the amendment and original text), but we relate those and other texts and amendments of the work implicitly by relating each of them to the entity we call the work.

Relating expressions of a work indirectly by relating each expression to the work that it realizes is often the most efficient means of grouping related expressions. In effect, the name given to the work serves as the name for the entire set or group of expressions that are realizations of the same intellectual or artistic creation. It is the entity defined as work, therefore, that provides this grouping capability.

In Akoma Ntoso [Barabucci et al. 2010a] the identification section classifies the document informing the semantic tools that the document is the manifestation of a certain abstract work, allowing them to distinguish between different versions of the same work. See 2.4.2.1. for a complete explanation of how Akoma Ntoso applies the $FRBR_{00}$ model to legal resources.

2.3.2. FRBR Classification: Expression

Expression is the intellectual (or artistic) realization of a work in the form of text, notation, sound, image, object, movement, or any combination of such forms.

An expression is the specific intellectual or artistic form that a work takes each time it is "realized." Expression includes, for example, the specific words, sentences, paragraphs, etc. that result from the realization of a work in the form of a text, or the particular notes, phrasing, etc. resulting from the realization of a musical work. The boundaries of the entity expression are defined, however, so as to exclude aspects of physical form, such as typeface and page layout, that are not integral to the intellectual or artistic realization of the work as such.

Since the form of expression is an inherent characteristic of the expression, any change in form (e.g., from text to spoken word) results in a new expression. Similarly, changes in the intellectual conventions or instruments that are employed to express a work (e.g., translation from one language to another) result in the production of a new expression. Strictly speaking, any change in intellectual or artistic content constitutes a change in expression. Thus, if a text is revised or modified, the resulting expression is considered to be a new expression, no matter how minor the modification may be. For example:

w1 The song “I Can’t Live With You” by Queen

e1 The original studio recording

e2 the 1997 "Rocks" retake

In the legal field, an expression identifies the exact combination of words of a legal text. Any modification of the text gives birth to a new expression. Please note that, since differences in page layout do not count, this is not the level where the "official version" of the law is identified. This function pertains to the lower level (manifestation).

On a practical level, the degree to which bibliographic distinctions are made between variant expressions of a work will depend to some extent on the nature of the work itself, and on the anticipated needs of users. Differences in form of expression (e.g., from musical notation to recorded sound) will normally be reflected in the bibliographic record, no matter what the nature of the work itself may be. Variant expressions in the same form (e.g., revised versions of a text) will often be indirectly identified by the data used to identify the manifestation in which the expression is embodied (e.g., an edition statement, a software's version). Variations that would be evident only from a more detailed analysis and comparison of expressions (e.g., variations between several of the early texts of Shakespeare's Hamlet) would normally be reflected in the data only if the nature or stature of the work warranted such analysis, and only if it was anticipated that the distinction would be important to users.

In the legal field, differences in laws and judicial decisions would be tracked with reference to the deliberations (and registry entry) which caused those differences. Different contributions in doctrine would be reflected in editorial data. The anticipated needs of users could represent the legal relevance of a legal document's version, which does not say much since everything that is formal in the legal setting is also relevant. Small details and minimum differences between legal documents could always become relevant if the document's formal integrity (and thus its validity) is at stake.

Defining expression as an entity in the model gives a means of reflecting the distinctions that may exist between one realization and another of the same work. It also allows to draw relationships between specific expressions of a work. The entity called expression can be used to identify, for example, the specific version of the law on which a piece of legal doctrine is based, or the specific version of the commentary used for a citation or comparison.

The entity defined as expression can be also used to indicate that the content embodied in one manifestation is in fact the same as that embodied in another manifestation. If two manifestations embody the same content, even though the physical embodiment may differ and differing attributes of the manifestations may obscure the fact that the content is the same in both (i.e. if two books contain the same piece of legal doctrine), we can make the common link through the entity defined as expression.

In Akoma Ntoso, the expression layer represents the original legal text. Thus, any modelling done to further qualify the legal text through ITs represents a manifestation of that expression.

2.3.3. FRBR Classification: Manifestation

The third entity defined in the model is manifestation: the physical embodiment of an expression of a work.

The entity defined as manifestation includes a wide range of materials, including manuscripts, books, periodicals, maps, posters, films, CD-ROMs, DVDs, etc. As an entity, manifestation represents all the physical objects that bear the same characteristics, in respect to both intellectual content and physical form.

When a work is realized, the resulting expression of the work may be physically embodied on or in a medium, such as paper, audio tape, video tape, canvas, plaster, etc. That physical embodiment constitutes a manifestation of the work. In some cases there may be only a single physical exemplar produced of that manifestation of the work (e.g., an author's manuscript, a tape recorded for an oral history archive, an original oil painting, etc.). In other cases there are multiple copies produced in order to facilitate public dissemination or distribution. Whether the scope of production is broad (e.g., in the case of publication, etc.) or limited (e.g., in the case of copies made for private study, etc.), the set of copies produced in each case constitutes a manifestation. All copies produced that form part of the same set are considered to be copies of the same manifestation.

The boundaries between one manifestation and another are drawn on the basis of both intellectual content and physical form. When the production process involves changes in physical form the resulting product is considered a new manifestation. Changes in physical form include changes affecting display characteristics (e.g., a change in typeface, size of font, page layout, etc.), changes in physical medium (e.g., a change from paper to microfilm as the medium of conveyance), and changes in the container (e.g., a change from cassette to cartridge as the container for a tape). Where the production process involves a publisher, producer, distributor, etc., and there are changes signaled in the product that are related to publication, marketing, etc. (e.g., a change in publisher, repackaging, etc.), the resulting product may be considered a new manifestation. Whenever the production process involves modifications, additions, deletions, etc. that affect the legal content, the result is a new manifestation embodying a new expression of the work.

w1 Ridley Scott's "Blade Runner" motion picture

e1 The original version

m1 The film distributed in theaters in 1982

m2 The 10th anniversary VHS version

e2 The director's cut

m1 The DVD version released in 1997

m2 A .avi file ripped from the DVD version

…

w1 The song "I Can't Live With You" by Queen

e1 The original studio recording

m1 the fourth song in the "Innuendo" music tape m2 the fourth track in the “Innuendo” CD

e2 the 1997 “Rocks” retake

m1 the fifth track of the “Queen Rocks” CD

m2 the MP3 available at amazon.co.uk

In the legal field, manifestation is useful under different aspects. It allows to define the legally binding version of a legal text. For laws, the version of the Official law Gazette can be distinguished from any reproduction of the text contained in it (which is not legally binding). For judgements, the original document and their legal copies (for the parties) can in this way be distinguished from their (even wordly) reproduction in case-law books, their online versions, and so on. For legal doctrine, the general considerations for books apply.

w1 Civil Code (Art. 45)

e1 Original Version

m1 The Special Edition of the Official Gazette n. 79 of April 4th, 1942 (Art. 45)

m2 The book “Civil Code” published by Mondadori, ed. 1952 (Art. 45)

e2 Version modified by law 151 of 1975 (Art. 45)

m1 The Official gazette n. 135 of May 23rd, 1975 (Art. 45)

m2 The online version available at www.altalex.com (Art. 45)

…

w1 Decision 1348/1998 of the Court of Cassation

e1 The original version

m1 The copy for the parties

m2 A photocopy made by one of the parties for his personal records.

e2 the summary in the collection of cassation court

m1 The summary in the paragraph “1348/1998” of the Cassation Court case-law collection

m2 The online version of the page in DeJure online database

Changes that occur deliberately or even inadvertently in the production process that affect the copies result, strictly speaking, in a new manifestation. Changes that occur to an individual copy after the production process is complete (e.g., the loss of a page, rebinding, or an error in the copy of the MP3 file) are not considered to result in a new manifestation. That copy is simply considered to be an exemplar (or item) of the manifestation that deviates from the copy as produced. In the legal field, this can be used to represent the system of "official copies" of legal acts: these are all considered as items of the same manifestation.

Defining manifestation as an entity allows naming and describing the complete set of items that result from a single act of physical embodiment or production. The entity serves to describe the shared characteristics of copies of a particular publication, edition, release, etc., as well as to describe unique productions such as manuscripts, original oil paintings, etc. In the legal field, it is useful to identify a unit of text as a speech act, and to relate it to the corresponding legal concept(s).

Through the manifestation entity it is possible to describe the physical characteristics of a set of items and the characteristics associated with the production and distribution of that set of items that may be important factors in enabling users to choose a manifestation appropriate to their physical needs and constraints, and to identify and acquire a copy of that manifestation. It also allows drawing relationships between specific manifestations of a work. We can use the relationships between manifestations to identify, for example, the specific publication that was used to create a micro-reproduction. In the legal field, it is fundamental to identify the legally binding version of a work.

In Akoma Ntoso, the manifestation layer represents the level of the marked-up XML document. The general schema employed by Akoma Ntoso documents to express assertion is:

> "the author of a manifestation asserts on the manifestation date that the author of the corresponding expression asserts on the expression date in a particular context that subject does predicate on object" [Barabucci et al. 2010a], p. 144

2.3.4. FRBR Classification: Item

The fourth entity defined in the model is item: a single exemplar of a manifestation.

An item is a concrete entity, constituted by one or more physical objects (i.e., an encyclopedia is a single item even though it is made of several volumes). In terms of intellectual content and physical form, an item exemplifying a manifestation is normally the same as the manifestation itself. However, variations may occur from one item to another, even when the items exemplify the same manifestation, where those variations are the result of actions external to the intent of the producer of the manifestation (e.g., damage occurring after the item was produced, binding performed by a library, etc.). Following is an exemplification of all the four layers for the "Blade Runner" example:

w1 Ridley Scott's "Blade Runner" motion picture

e1 The original version

m1 The VHS 10^{th} anniversary edition

i1 The copy of the VHS sold on Ebay with id EPID3056531

i2 The copy of the VHS in Harrison Ford's bookshelf

In the legal field, it is particularly useful for legal sources such as contracts where the differences between single specimens may be relevant, or where it may be important to refer to a specific copy (i.e. consumer's copy) of a contract, as explained in the example:

w1 the telecom contract for telephone line

e1 the 2005 "Tutto Compreso" consumer contract

m1 The contract printed in july 9^{th}, 2005

i1 the copy held by the consumer Mario Rossi

Defining item as an entity enables to separately identify individual copies of a manifestation, and to describe those characteristics that are unique to that particular copy and that pertain to transactions such as circulation, etc. involving that copy. It also allows drawing relationships between individual copies of manifestations.

In the legal field, this is useful everytime when bring specific means of proof (not only a document but also an object, a probe, etc.) Through the "item" element we can distinguish the single object even if they are used as mere tokens to explain facts or circumstances through deduction. The factors (or circumstances) that have been proved may lay on the item layer or at an upper layer. For example, if a single toy is acquired by the court to prove a fault in the production, it is important to indicate whether the results of the probe refer to the single toy (item), to the production batch (manifestation) or to the toy's design (expression). As said in the beginning of the section, while work, expression, and manifestation are intentional objects (they exist only as the object of one's thoughts and communication acts), an item is a physical object. Items stored on a computer, however, can be easily copied to another location, resulting in another item, but still an instance of the same manifestation. This makes it impossible, in principle, to add the metadata about the item directly on it. On the Internet only the uniform resource locator (URL) is an item-specific datum. The item level is therefore not very relevant to XML standards.

2.4. Semantic Web Projects Related to the Legal Field

2.4.1. URNs and XML

2.4.1.1. CEN Metalex

CEN Metalex is defined as an Open XML interchange format for legal and legislative resources. Its goal (see [Boer et al. 2009]) is to standardize the way in which sources of law and references to sources of law are to be represented in XML. Being an interchange format, it acts as a lowest common denominator for other standards. It is therefore not intended to replace jurisdiction-specific standards and vendor-specific formats in the publications process, but rather to create a standardized view on legal documents for the purpose of information exchange and interoperability in the context of software development. MetaLex includes a mechanism for schema extension, adding metadata, cross referencing, constructing compound documents and a basic naming convention. The MetaLex schema is based on the best practices from – amongst others – the previous versions of the MetaLex schema, the Akoma Ntoso schema, and the Normeinrete schema. Other important sources of inspiration are LexDania, CHLexMOL, FORMEX, R4eGov, all available on the web. MetaLex also implements the distinctions made by the IFLA FRBR (see 2.3.).

MetaLex precribes what counts as a MetaLex metadata statement, how it is stored inside a MetaLex document, and what classes of entities and which predicates (properties) MetaLex distinguishes. The RDF ontology classifies the following entities:

– **Bibliographic entities**: the work, expression, manifestation, and item level, and content models;

– **Reference**: type of reference between bibliographic entities;

– **Activities**: actions and thematic links, and thematic roles of bibliographic entities in at least the actions creation, enactment, repeal;

– **Agent and competence**: the agents and institutional instruments (legislative power, etc.) used in legislative activity.

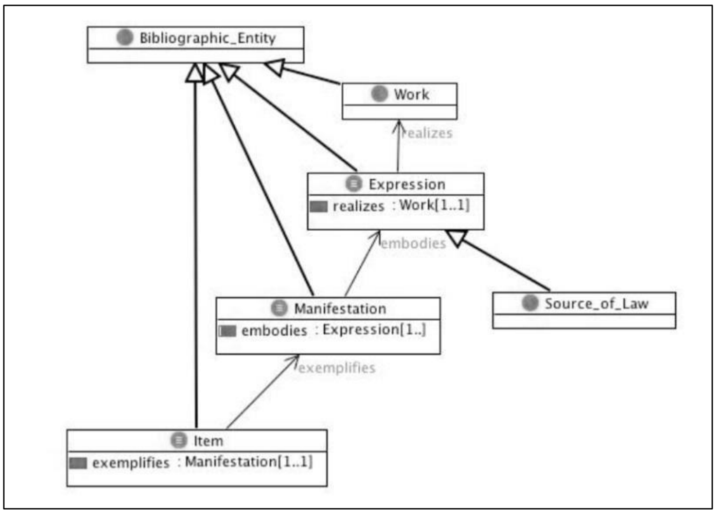


Fig. 2.6 – A taxonomy of bibliographic entities in MetaLex

MetaLex meta elements are used to embed metadata that can be stored in the form of RDF statements in RDF documents. Elements derived from the meta content model are carriers of RDFa attributes, and are therefore of RDFa statements. All entities are identified using URIs, and the ontology is extensible.

2.4.1.2. The NormeInRete Project (NiR)

The Normeinrete project (NiR) aims at fulfilling the citizens' right to acquire knowledge of legislation, while at the same time supporting Public Administration in efficiently managing the legislative documentation lifecycle. The project was promoted by the Italian Authority for Information Technology in Public administration (AIPA) and the Italian Ministry of Justice, and it comprehends by the following actions:

– A specialized portal for legislative documents retrieval;

– Standards to represent legally relevant data;

– Software distribution to support legislative document management and publishing;

– Specific courses among public administrations.

The system is based on a federation of legislative data bases developed with different platforms and it is built upon a co-operative technological architecture.

Normeinrete improves accessibility to legislation by providing a unique point of access to Italian and European Union legal documents published on different web sites through a specialized portal (formerly www.nir.it, now moved to www.normattiva.it after the renaming of the whole project as “NormAttiva”). The portal runs a search engine that operates homogeneously on distributed data sources. Its full-text search index is selectively built detecting only legislative documents. The standards have been issued as AIPA technical norms and published as regulatory acts in the Italian Official Journal. The definitions make use of IETF URNs[9] and XML standards[10]. Normeinrete supports the public administration also in the tasks related to law consolidation by providing standard definitions which allow to identify the norms in a distributed environment and to track their modifications, thus performing semi-automatic consolidation. The project also aims at creating a virtual space for knowledge sharing within the public administration community, through dedicated services such as e-learning tools and open source software. So far, more than 40 public institutions have taken part in the project and more than 140.000 documents have been indexed. The site supplies about 150.000 search sessions monthly.

Normeinrete introduced specific elements for URNs, which are used to represent the references to laws contained in other laws. The basic elements are: name of the promulgating authority, type of law, date, number, as well as a set of more detailed, accessory specifications. The adoption of a scheme based on URNs allows the automatic building of an hypertext according to a model similar to the DNS (Domain Name System) used to resolve the self-explaining web sites names into numerical HTTP addresses. This opportunity relies on the consideration that the natural language expressions used in the quotation of laws usually contain repetitive patterns, making references automatically detectable. The URN can then be built by combining data which are almost always included in the reference.

The XML representation of legislative documents improves effectiveness in managing, publishing and retrieving norms by electronic means (see [Barabucci et al. 2010b]). Normeinrete has defined a DTD (Document Type Definition) for the Italian legislation, considering the peculiarity of legislative documents and other significant useful information. Italian legislative and regulatory acts can be divided into three categories:

- Documents with a well-defined structure (state laws, regional laws, etc.);
- Partially structured documents (regulation act, decrees, etc.);
- Generic documents (enclosures, informal acts, etc.).

To avoid a proliferation of DTDs, it has been considered more convenient the definition of a single DTD containing many elements capable of representing all the types of documents.

[9] Circolare del 6 novembre 2001, n. AIPA/CR/35 - Assegnazione dei nomi uniformi ai documenti giuridici. Available at http://www.digitpa.gov.it/sites/default/files/normativa/Circ_AIPA_2001 _11_06_n_35.pdf

[10] Circolare 22 aprile 2002 n. AIPA/CR/40 - Formato per la rappresentazione elettronica dei provvedimenti normativi tramite il linguaggio di marcatura XML. Available at http://www.digitpa.gov.it/sites/default/files/normativa/Circ_AIPA_2002_04_22_n_40.pdf

Because the structure of Italian legislative documents may vary, the mark-up language is very complex and the resulting DTD has three different versions, containing the same set of elements to represent all kinds of documents, but relying on different constraints to distinguish them. Documents validated against strict rules are also valid against looser ones.

Mark-up must be carried out using only elements relevant to the kind of document. The DTD elements can be classified as follows:

- Structural elements, identifying the parts in which the document is structured (heading, preamble, articles, etc.);
- Special elements identifying meaningful parts of the text in the legal context (for instance references to other laws) or associating a formatted representation to text embedded relevant entities (institutions, dates, places);
- Elements containing metadata (for instance subject-matter classification, publication data, procedures to enact a bill, etc.).

2.4.2. The Akoma Ntoso Framework

The Akoma Ntoso (*Architecture for Knowledge-Oriented Management of African Normative Texts using Open Standards and Ontologies*) framework is a set of guidelines for e-Parliament services in a pan-African context. The framework was developed within the “Strengthening Parliaments’ Information Systems in Africa” project of the United Nations Department for Economics and Social Affairs (UN/DESA), a project aimed at empowering legislatures to better fulfill their democratic functions by using ICTs to increase the quality of parliamentary services, facilitate the work of parliamentarians and create new ways to promote the access of civil society to parliamentary processes.

Akoma Ntoso (see [Barabucci et al., 2010a]) focuses on information content and issues recommendations, technical policies and specifications for building and connecting Parliament information systems across Africa. In particular, its framework proposes an XML document schema providing sophisticated description possibilities for several parliamentary document types (including bills, acts and parliamentary records, etc.), therefore fostering easier implementation of parliamentary information systems and interoperability across African parliaments, ultimately allowing open access to this information.

The framework reaches three main objectives which are instrumental for the success of the overall project:

- To define a common standard for data interchange between parliaments;
- To define the specifications for a base document model on which parliamentary systems can be built;
- To define an easy mechanism for citation and cross referencing of data between parliaments.

The Akoma Ntoso framework aims at providing two basic types of interoperability: semantic interoperability (which means ensuring that the precise meaning of exchanged information is understandable by any person or application receiving the data) and technical interoperability (aimed at ensuring that all Akoma Ntoso-related applications, systems, interfaces are based on a shared core of technologies, languages and technical assumptions easing data interchange, data access and reuse of acquired competencies and tools). Akoma Ntoso ensures technical interoperability by enforcing the use of open standards and open document formats, based on the XML language.

The Akoma Ntoso metadata format is primarily focused on resource discovery and records management. The aim is to ensure that people searching the knowledge online have easy access to descriptions of many different resources. In this way, advanced search is possible as well as the generation of structured collections of legislative terms across disparate documents. The Akoma Ntoso metadata format is also designed to be extensible (see [Vitali, 1997]).

2.4.2.1. $FRBR_{00}$ and Legal Resources

[Lima et al. 2008] gives the following operational definitions of the main legal concepts:

- **Norm**, a rule of conduct issued by a competent authority and prescribing or regulating behavior among individuals and within society. Its form of expression may be the written or the spoken word, but it may also be visual or be based on usage and custom.
- **Normative provision**, any group of words or piece of writing expressing a norm or series of norms.
- **Normative document** or **act**, an officially published, legislative, written document through which a competent authority brings a norm into being.
- **Legal system**, a set of norms belonging by some criterion to a single system and related to one another in different ways, as by hierarchy (one norm having a higher or lower standing than another), generality (more specific or more general), time (issued before or after another norm), and modification (one norm modifying the other norm or getting modified by it).
- **Normative system**, the same legal system viewed in a diachronic perspective: its changes over time can be represented in its evolution as a series of snapshots or film-stills in succession. The sequence in the time of legal systems so captured called the normative system. [Palmirani 2005]

Akoma Ntoso classifies legal resources following the $FRBR_{00}$ model. In the model, represented in figure 2.7 from [Palmirani 2005], an original document (F4 Manifestation Singleton) is created through the “F27 Work Conception” and “F28 Expression Creation” events. The “F32 Carrier Production” event forms a “F19 Publication Work” which, in turn, produces various items. Following is a list of classes and instances which can be identified in a signed official normative document (the original signed act):

- **Manifestation singleton**: the original signed document as a physical object that carries an instance of Self Contained Expression;
- **Individual work**: the concepts associated with the signed act;
- **Self-contained expression**: the normative provision resulting from the legislative process.

Classes and instances which can be identified on a page of an official publication are classified as follows, distinguishing between the single entities (the result of the industrial process) and the relations to abstract legal resources:

- **Item**: the Official Gazette issue (physical object);

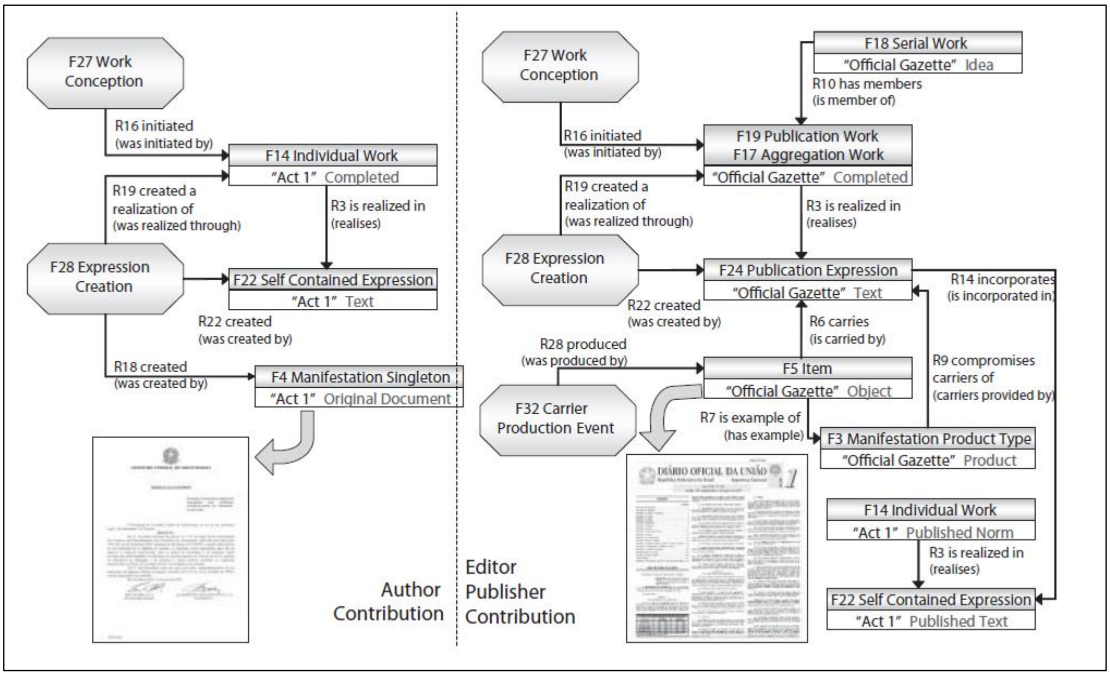


Fig. 2.7 – Work Conception, Expression Creation and Carrier Production Events in [Palmirani 2005]

- **Manifestation product type**: the publication product type with a given title, date and author;
- **Serial work**: the periodical with a given title;
- **Publication work**: the concepts associated with the official publication with a given date and author;

– **Publication expression**: complete layout and content provided by a publisher;

– **Self-contained expression**: the normative provisions published;

– **Individual work**: the concepts associated with the published acts.

In 5.3.2. it is shown how the same model is used to represent the lifecycle of a legal norm.

2.4.2.2. Judgement Structure

The judgement in Akoma Ntoso is a particular type of document modelling for detecting the main significant parts of the precedent document: *header* for capturing the main information (parties, *coram*, neutral citation, document numbers and identification information); *body* for representing the main part of the judgement, including the decision; *conclusion* for detecting the signatures.

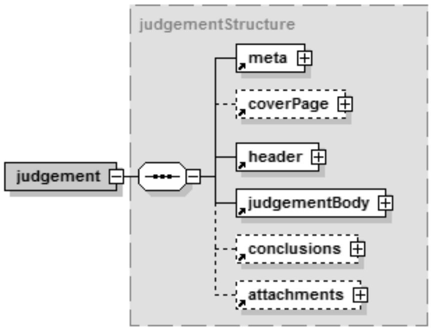


Fig. 2.8 – Judgment main structure in Akoma Ntoso

The body is divided into four main blocks:

– The **introduction**, where usually (especially in common law decisions) the story of the trial is introduced;

– The **background**, dedicated to the description of the facts;

– The **motivation**, where the judge introduces the arguments supporting his decision;

– The **decision**, where the judge gives the final outcome.

This division is fundamental for detecting facts and factors from the background: in the motivation, to detect arguments and counterarguments; in the decision, to detect the conclusion of the legal argumentation process. Those qualified fragments of text should be annotated by legal experts with the help of a tool for linking text, metadata and ontology classes (see 5.2.1.).

2.4.2.3. Metadata of judgements

The metadata of the judgements are divided in different blocks:

– **Descriptive** metadata, tracking general data of the judgement such as the date of publication, the number of the case, the natural citation, the names of the judges, the jurisdiction, the level of the judgement, the nature of the case, the type of court, the parties, the lawyers, and so on;

– **Classification** metadata, concerning the matter of the case, together with the reportable or not reportable case-base. These metadata represent a filtering station of reportable judgements, following the common law tradition that underlines the cases producing a new rule of law;

– **Lifecycle** metadata, containing the history of the document, which is useful for versioning;

- **Workflow** metadata, tracking each step of of the document production process. Since multi-annotation of the same fragment of text is allowed, each actor in the workflow chain can annotate the document with his/her specific metadata;

- **Reference** metadata, remarking all documents citing/cited by the judgement or links all documents which are logically connected to the judgement;

- **Semantic** metadata, the annotation and classification of the text under the legal point of view, particularly relevant in the decision’s groundings;

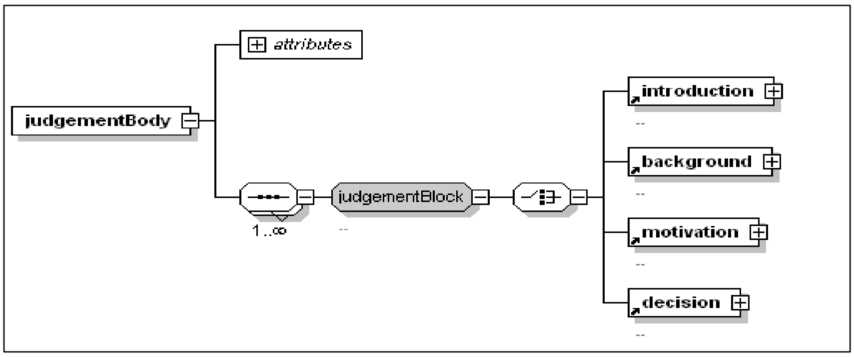


Fig. 2.9 – Judgment Body sub-elements in Akoma Ntoso

- **Ontology** metadata, a definition of criteria to link the fragment of text to general classes of the computational ontology.

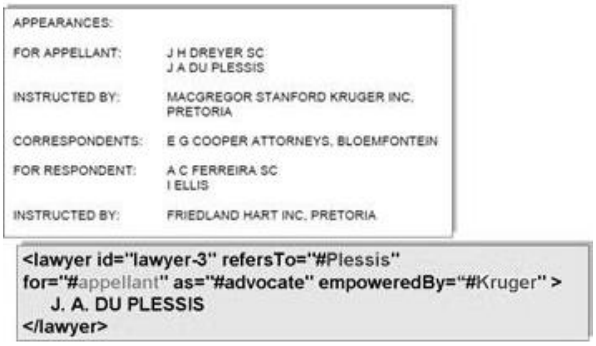


Fig. 2.10 – Actor qualification.

Using these metadata it is possible to annotate very specific knowledge. In the following fragment of text, it is necessary to capture the role of each person involved in the trial: in the example of Figure 2.10, Mr. Du Plessis is a lawyer, with the role of advocate of the appellant, instructed by the Kruger Inc. It is possible to annotate these information with XML to allow complex queries such as: “give me all the judgements where Du Plessis is playing the role of instructor of the appellant on behalf of a third Inc. company”.

2.4.2.4. Qualification of the citations in the judgement

Each judgement citation can be qualified using Shepard's method that permits to understand which references are in favor of the current judgement argumentation and which are not. The list of qualifications includes:

- **Support**, meaning that the cited judgement supports the current decision;
- **IsAnalogTo**, meaning that the current case-law is analogue to a cited precedent;
- **Distinguished**, meaning that the current precedent is distinguished from the cited case-law;
- **Overrules,** particularly important since it detects the case-law whose rule of law the judge intends to overrule (through a citation in the judgement's interpretation).

This qualification mechanism helps to reinforce the main arguments used by the judge to provide evidences and parameters (e.g. returning a list of the cited case-law with the role played in the argumentation).

2.4.2.5. Rule of law, stare decisis and ratio decidendi, obiter dicta

One of the main tasks in common law judgements is to define a rule of law fixing the pattern for the similar future cases. This monotonic mechanism is called *stare decisis* and guarantees the equal application of justice to comparable cases. The *stare decisis* is applied only to a particular and relevant part of the decision called *ratio decidendi*, excluding accessories arguments called *obiter dicta*. The research conducted on the ontology framework (see Chapter 3) reveals the importance of marking up those relevant and meaningful parts of the text. By marking up the ratio decidendi text and using this information in combination with the Shepard's qualification method for cited cases-law, it is possible to provide richer information to the argumentation engine devoted to the legal reasoning. For these reasons the present research will define new metadata in the analysis block of Akoma Ntoso in order to qualify also the ratio decidendi.

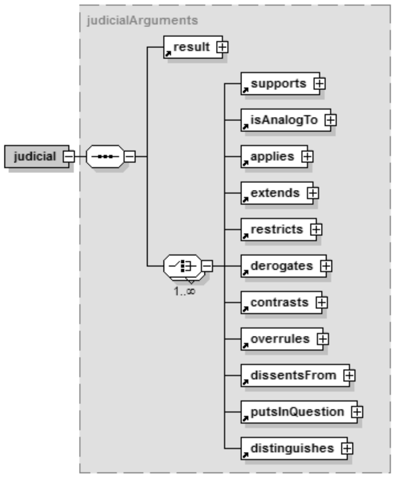


Figure 2.11: Shepard's method qualification

2.4.3. Foundational ontologies

2.4.3.1. Functional Ontology of Law (FOLaw)

Several ontologies for the legal domains were developed in the last two decades. Core ontologies were conceived in the mid-90s to consolidate the insights acquired in modelling legal domain knowledge and to provide a conceptual framework for developing ontologies. Among the first projects in that direction is the "Functional Ontology of Law (FOLaw, [Valente 1995]).

FOLaw represents a legal-sociological view rather than a perspective from the law itself, and therefore it lacks the abstract, core concepts that make up law. Being a functional ontology, the roles that the legal system plays in society are taken as point of departure. FOLaw identifies the dependencies between the types of knowledge, which indicate the roles that types of knowledge play in the reasoning. The following six types of knowledge are distinguished in [Breukers 2004] and represented in Figure 2.12.

- **Normative knowledge**, containing deontic operators decompiled into a binary value;
- **Meta-legal knowledge**, used to solve conflicts between individually applicable norms;

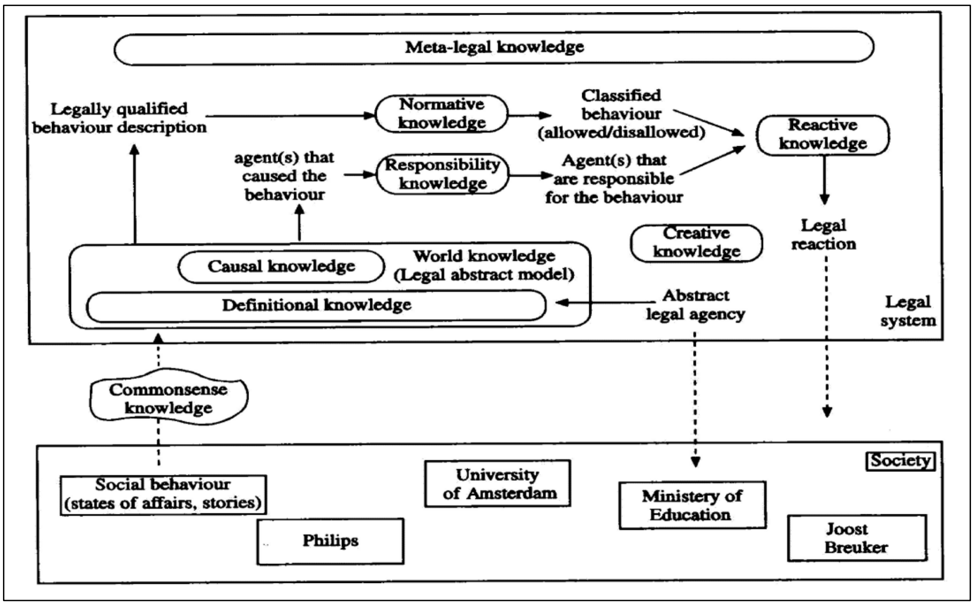


Figure 2.12: FOLaw, a functional ontology for law as represented in [Breukers, 2004].

- **World knowledge**, acting as a filter between the legal system and the actual events that happen in some jurisdiction, and including common sense-terms used to describe them and the causal connections between them.
- **Responsibility knowledge**, used to assign or to limit the responsibility of an agent over a given (disallowed) state of affairs;
- **Reactive knowledge**, specifying which reward or sanction should be taken;
- **Creative knowledge**, used to represent the constitutive powers of the legislator.

The framework of FOLaw was used as a lead for research on responsibility [Lehmann 2003], as the basis for practical applications and architectures for legal reasoning such as ON-LINE [Valente et al. 1999], and in the CLIME project aimed a t the construction of a legal information server [Winkels et al. 2002]. These applications show that FOLaw is more a reasoning structure that model the way legal cases are argued for, rather than an ontology. Its dependencies bring, in fact, similarities to inference structures: representations of problem solving methods that specify the problem decomposition and its dependencies (see [Breuker and Van De Velde 1994] and [Schreiber et al. 2000]).

2.4.3.2. Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE)

As its name suggests, the Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) has a clear cognitive focus, in the sense that it aims at capturing the ontological categories underlying natural language and common sense. The ontology is not committed to a metaphysics about the intrinsic nature of the world: its categories are rather thought of as cognitive artifacts ultimately depending on human perception, cultural imprints and social conventions (a sort of "cognitive" metaphysics, similar to Searle's notion of "deep background" [Searle 1983], which represents the set of skills, tendencies and habits shared by humans because of their peculiar biological make up, and their evolved ability to interact with their ecological niches). As a consequence, the categories of DOLCE do not claim any special robustness against the state of the art in scientific knowledge: they are just descriptive notions that assist in making explicit conceptualizations which were already formed, without providing a prescriptive framework to conceptualize entities. DOLCE is an ontology of particulars, in the sense that its domain of discourse is restricted to entities which have no instances. Of course, universals (such as predicates and properties) do appear in the ontology, insofar they are used to organize and characterize them: simply, since they are not in the domain of discourse, they are not themselves subject to being organized and characterized (i.e. through metaproperties).

An important choice made in DOLCE is the so-called multiplicative approach: entities that are different (because they are ascribed incompatible essential properties) can be co-located in the same space-time. The example presented in [Masolo et al. 2002] is that of the vase and the amount of clay: necessarily, the vase does not survive a radical change in shape or topology, while, necessarily, the amount of clay does. Therefore the two things must be different, yet co-located.

The taxonomy of the basic categories of particulars assumed in DOLCE is depicted in Figure 2.13. They are considered as rigid properties, according to the OntoClean methodology [Guarino and Welty 2002] that stresses the importance of focusing on these properties first.

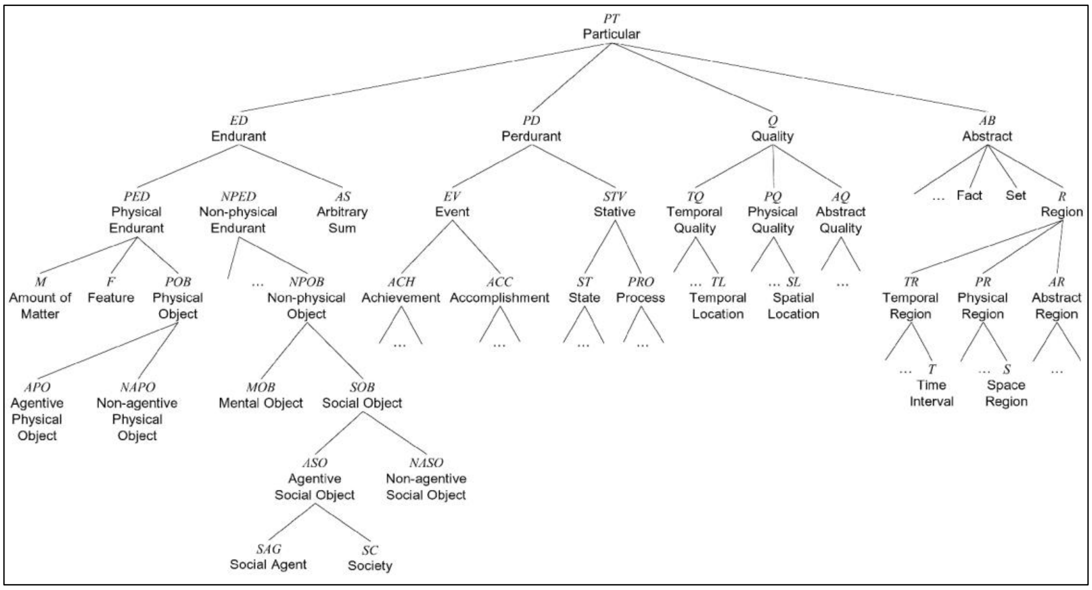


Fig. 2.13: Taxonomy of DOLCE basic categories from[Masolo et al. 2002].

DOLCE is based on a fundamental distinction between endurants and perdurants, based on their behavior in time:

- **Endurants** are wholly present (i.e., all their proper parts are present) at any time they are present. Endurants can "genuinely" change in time, in the sense that the very same endurant as a whole can have incompatible properties at different times;

- **Perdurants,** instead, extend in time by accumulating different temporal parts so that, at any time they are present, they are only partially present, in the sense that some of their proper temporal parts (e.g., their previous or future phases) may be not present. Perdurants cannot change in time in a strict sense, since none of their parts keeps its identity in time.

In DOLCE, the main relation between endurants and perdurants is that of participation: an endurant "lives" in time by participating in some perdurant(s). For example, a person, which is an endurant, may participate in a discussion, which is a perdurant. A person's life is also a perdurant, in which a person participates throughout its all duration.

In the DOLCE ontology, space and time locations are considered as individual qualities like colors, weights, etc. Their corresponding sensorial regions are called spatial (temporal) regions. For example, the spatial location of a physical object belongs to the quality type space, and its sensorial region is in the geometric space. This allows an approach that remains neutral about the properties of the geometric/temporal space adopted: for instance, one is free to adopt linear, branching, or even circular time.

Two kinds of quality inherence are distinguished in DOLCE: direct and indirect inherence. The main reason for this choice comes from the symmetric behavior of perdurants and endurants with respect to their temporal and spatial locations: perdurants have a well-defined temporal location, while their spatial location seems to come indirectly from the spatial location of their participants; vice versa, most endurants have a clear spatial location, while their temporal location comes indirectly from the that of the perdurants they participate in.

According to the general methodology introduced in [Gangemi et al. 2001], the DOLCE backbone properties rely on a set of basic primitive relations, suitable to characterize the ontological commitments as neutrally as possible. According to [Masolo 2002], these relations should be:

- General enough to be applied to multiple domains;

- Such that they do not rest on questionable ontological assumptions about the ontological nature of their arguments;

- Sufficiently intuitive and well-studied in the philosophical literature;

- Hold as soon as their relata are given, without mediating additional entities.

Following are the axioms constraining the arguments of primitive relations;

- **Parthood**: *x* is part of *y*;

– **Temporary Parthood**: *x* is part of *y* during *t*;

– **Constitution**: *x* constitutes *y* during *t*;

– **Participation**: *x* participates in *y* during *t*;

– **Quality**: *x* is a quality of *y*;

– **Sensioral Region**: *x* is the sensorial region of *y* (during *t*).

2.4.4. Core Ontologies

2.4.4.1. LRI-Core

LRI-Core is a common sense-based foundational ontology that focus around six major categories introduced in [Breuker and Hoekstra 2004]:

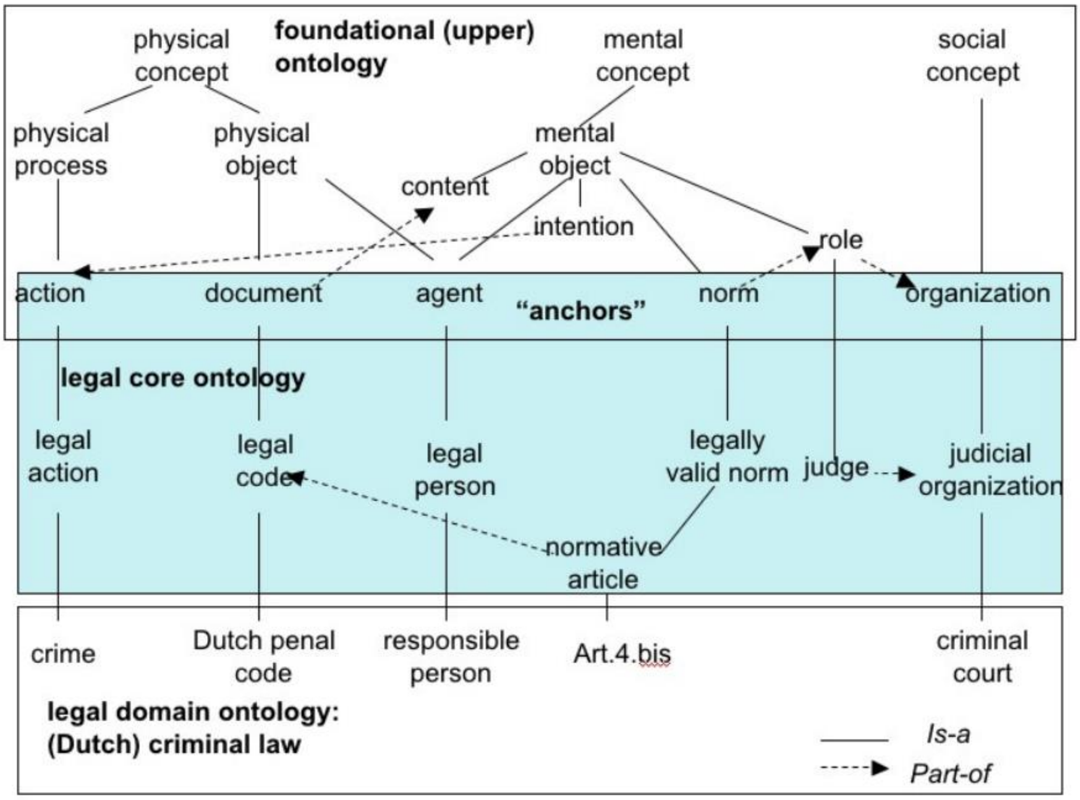


Figure 2.14: LRI-Core layers: foundational and legal core share *anchors* (high level concepts typical for law) in [Breuker and Hoekstra 2004].

– **Physical concepts**, divided between objects and processes;

– **Mental concepts**, also divided between objects and processes;

– **Roles**, representing notions about social behavior;

– **Abstract concepts,** for simple proto-mathematicfal ideas;

- **Occurrences**, which are not by themselves part of the ontology but which can be talked about through specific terms.

Figure 2.14 shows the layered structure of LRI-Core which contains two levels, the most abstract being a “shallow”, foundational ontology that covers concepts from physical, abstract, mental worlds and roles. Those properties are inherited by the core ontology, which contains more specific classification of the various worlds. In Figure 2.14 a subdivision of the foundational classes in the legal core ontology is shown. The last layer is the domain ontology, containing the specific norms and other entities involved.

LRI-Core constituted an important advancement in multi-layer ontology modelling for the legal domain. Its insights constitute the basis for successive projects such as LKIF (see 2.4.5.)

2.4.4.2. Core Legal Ontology (CLO)

The Core Legal Ontology [Gangemi et al. 2004] is developed on top of DOLCE and Descriptions and Situations ontologies [Gangemi and Mika 2003] within the DOLCE+ library. It has been used to support the definition of legal domain ontologies [Sagri et al. 2004], and of a juridical wordnet [Sagri 2003]. Its two pillars are the stratification and reification principles:

- According to the **stratification principle**, CLO classifies and relates the heterogeneous entities in the legal domain distinguishing between the physical, cognitive, social, and legal worlds (or layers);

- According to the **reification principle**, CLO represents intensional specifications like norms, contracts, subjects, and normative texts in a specific domain through their extensional realizations like cases, contract executions, agents, physical documents.

An important implication of the reification principle is that the link between the specification and the realization is achieved through two patterns, called Description and Situation:

- The **Description** pattern models the structure of an intensional specification, as composed of its concepts and their internal dependencies. For example, the structure of a norm (a legal description) employs that pattern.

- The **Situation** pattern models the matching between a description and its extensional realization, which can be described as the configuration of a set of entities according to the structure of a description.

A legal application of these patterns is contained in [Gangemi 2007] and takes as an example the dependencies among the rules in a contract, when they can be matched to a legal case (a legal situation or fact) or a contract execution. The matching is typically performed when checking if each entity in a legal fact is compliant to a concept in a legal description. As will be seen in Chapter 3, the reification principle has been taken into consideration in the present research.

2.4.5. The legal Knowledge Interchange Format (LKIF)

The Legal Knowledge Interchange Format (LKIF) is an XML application developed in the European Estrella project (see [Hoekstra et al. 2007, Boer et al. 2008]) with the goal of establishing an open standard for exchanging formal models of the law, suitable for use in legal knowledge-based systems. The Estrella consortium includes a number of companies experienced in building legal decision support systems, academic partners from the AI & Law field, and some public bodies. LKIF aims at encoding in XML four kinds of legal knowledge: arguments, rules, ontologies and cases. The application is intended to serve two main purposes:

- As a reusable and extensible **core ontology**, **application programmer interface** (API), and **inference engine specification** for legal decision support systems, knowledge management systems, and argumentation support systems;
- As an **interchange format** for existing (proprietary) legal knowledge representation languages.

The requirements for LKIF are derived from several sources:

- A **survey of research** on computational models of legal reasoning and argumentation, from the field of Artificial Intelligence and Law;
- An **analysis** of the business requirements articulated by the participating vendors, and from the logical reconstructions of the logics used by these vendors;
- **Feedback** and **comments** by the participating user organisations and members of the observatory board.

LKIF is a knowledge representation language for legal arguments, rules, terminological axioms, and cases. It can be characterized as an ontology of law for the Semantic Web, and as a knowledge representation language specifically suitable for legal reasoning in its own right. On the language level LKIF combines existing Semantic Web technology (RDF and OWL) – and a new LKIF-Rules language extending RDF and OWL for dealing with presumptive inferences. The Rules language has an argumentation-theoretic semantics: its semantics is defined in terms of argumentation schemes.

2.4.5.1. The LKIF Ontology

The LKIF ontology is a standard OWL ontology, based on Description Logic (DL), and can be also used separately from the LKIF-Rules in Semantic Web applications. The main purpose of the ontology is to constrain the use of terminology in LKIF applications, and the ontology is intended to be the part of the knowledge representation to be reused outside its original context, although the applicability of this doctrine in law generates a lot of discussion: as a fact, issues of terminology often become subject of legal argument. Important in legal knowledge representation are the concepts of obligation and permission. The deontic notions are underconstrained to accommodate differences of opinion on their interpretation, and play a considerably less central role in LKIF than in legal knowledge representation in general.

The LKIF ontology explcitly distinguishes physical, mental, social, and abstract world, and underlines the role of conventional metaphors based on the physical world in the construction of the mental, social, and abstract world. The strong focus of the LKIF ontology towards intentional entities is much different from metaphysically inclined top ontologies such as SUMO (see [Niles, Pease, 2001]), in which intention plays a less central role, but shows similarities with for instance the DOLCE ontology (see 2.4.3.2.) and the distinction set in [Dennett, 1989] between intentional, design and physical stance . The intentional stance introduces a model of the mind to account for changes brought about by agents. The agent acts, i.e. he initiates processes – physical, social, or mental – that bring about changes that are intended. From here it is only a small step to recognizing that the agent can also perform physical "formal" acts (e.g. signing something) to effect institutional change by communicating to others one's intention to make that change. This discourse can be strictly related to Searle's performative (see 1.1.9.). In DOLCE the distinction between agentive and non-agentive, which has a similar impact, is relatively prominent. The main criticism of DOLCE is that it is rather a representation of the terms used for describing knowledge, than a representation of knowledge itself.

One usage of LKIF is to use it as a basis for domain-specific ontologies. Note that the ontology consists of terminological axioms: all claims can be considered defeasible in law, but that the proposition represented by the claim terminologically entails some other proposition is not. An ontology is not falsifiable: it is an agreement about use of terminology. Because the OWL axioms are in principle not falsifiable they should be used with due care (i.e. to avoid giving rise to inconsistencies).

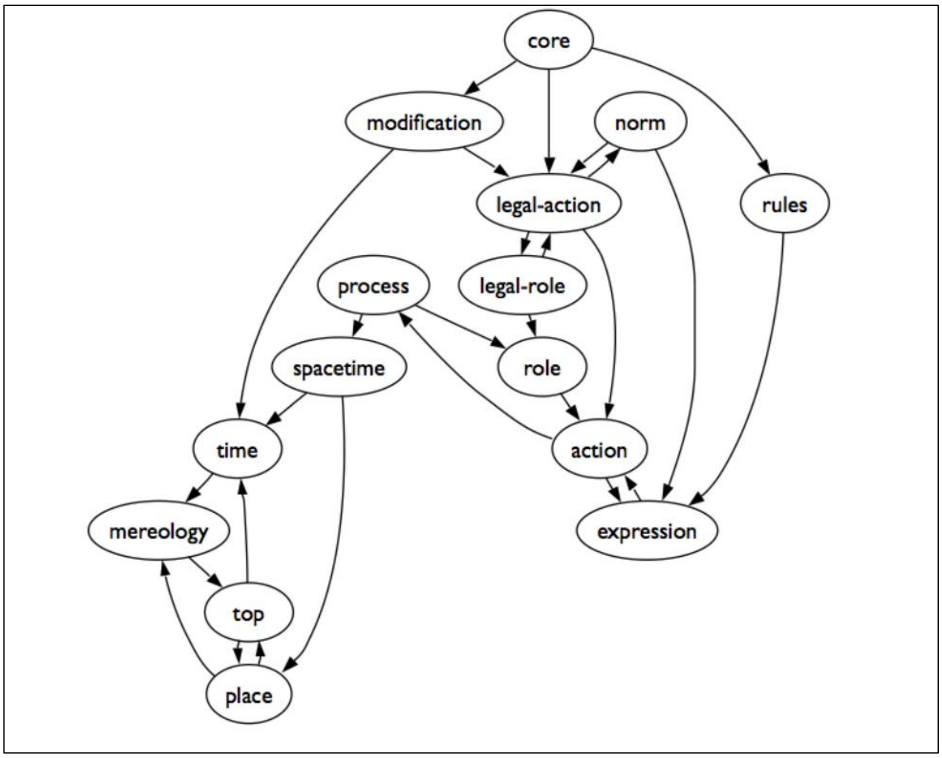


Figure 2.15: Dependencies between LKIF-Core modules in [Boer et al. 2007]

2.4.6. Rule languages

2.4.6.1. Prolog

Prolog is a general purpose logic programming language associated with artificial intelligence and computational linguistics. The language was first conceived by Alain Colmerauer in Marseille in the early 1970s [Colmerauer et al. 1972]. Its roots are in first-order logic, and unlike many other programming languages, Prolog is declarative: the program logic is expressed in terms of relations, represented as facts and rules. Prolog programs describe relations, defined by means of clauses, restricted to Horn clauses. Hence there are two types of clauses: facts (clauses without bodies) and rules (clauses with body). Execution of a Prolog program is initiated by the user's posting of a single goal, called the query. Logically, the Prolog engine tries to find a resolution refutation of the negated query. The resolution method used by Prolog is called SLD resolution.

Besides being not a project in the legal field, it is impossible not to cite Prolog when talking about rules languages as it constitutes the first programming language based on rules, therefore the first argumentation framework ever developed in ITs. Also for this reason, in the present thesis this language will be recalled every now and then by other computational architectures.

2.4.6.2. LKIF Rules

LKIF Rules semantics is based on the notion of defeasible rules as a type of argumentation scheme, particularly an argumentation scheme for each conclusion of the rule. Applying an LKIF rule means instantiating an LKIF argument from one of these schemes.

Argumentation schemes generalize the concept of an inference rule to cover plausible but non-deductive forms of argument. The semantics of LKIF rules , as specified by [Gordon 2008], is based on concepts also found in Carneades (see 4.2.). The issue of “plausible inference” is obviously relevant to any domain of knowledge, but it is particularly important in law because of the role played by explicit dialectical considerations in legal procedure. Argumentation schemes can also be used to fill the gap between a *prima facie* interpretation of the law, ignoring dialectical and procedural considerations, and the ways it is operationalised in specific administrative, and adversarial settings. In this case the same text would map to different interpretations in the form of LKIF rules, possibly in addition to a terminological interpretation. The concept of burden of proof in an adversarial setting provides good examples: for instance, anyone who kills a man is guilty of murder, unless he proves that he was acting for self-defense. Logically speaking the unless could be easily replaced with a (but) not, and turned into a terminological axiom, but by doing so you fail to take account of implied burden of proof: it is up to the plaintiff to argue that the killing was done by the defendant, and up to the defendant to argue it was made in self-defense. It is valid to argue that a man committed a murder because he killed another man, even if the argument may be defeated by arguments made by the counterparty. This aspect of burden of proof will be examined in 4.2.1.2.

The specification of the LKIF-Rules language can be found in [Boer et al. 2007]. In the present thesis, rules in LKIF-Rules will be presented mostly in XML source code (a clear yet verbose language with nested tags containing the name of the element: for example, *individual constants* are introduced with `<rel>`, *individual variables* with `<var>`, *connectives* with `<and>`, `<or>`, and so on), and occasionally in *s-expressions* syntax.

2.4.6.3. LegalRuleML

AI&Law systems (such as legal argumentation systems) should rely on open and shared standards. With that in mind, in the early 2000s the research community joined the efforts towards the definition of a standard for the syntax of legal rule extending LKIF-Rules with a modelling of temporal parameters, giving birth to the LKIF++ language [Palmirani et al. 2011c]. Soon realizing that a standard in syntactic representation of norms would require a shared rule language to be built from scratch, the OASIS consortium started the development of a brand new syntax for legal rules, explicitly relying on the acquired standards in the underlying layers of the Semantic Web cake.

A Rule Markup Language (RuleML) [Boley 2006] is a modular, interchangeable rule specification standard to express both forward (bottom-up) and backward (top-down) rules for deduction, reaction, rewriting, and further inferential-transformational tasks. It is defined by the Rule Markup Initiative [Boley and Tabet 2000], an open network formed to develop a canonical Web language for rule serialization using XML and for transformation from different rule standards/systems. The language family of RuleML covers the entire rule spectrum, from derivation rules to reaction rules including rule-based event processing and messaging, as well as verification and transformation rules.

LegalRuleML [Palmirani et al. 2011c] is an extension of RuleML, an XML based language for the representation of legal rules using formal semantics described in [Lee & Sohn 2003]. LegalRuleML introduces features which are fundamental for modelling legal rules:

– Isomorphism;
– Defeasible logics;
– Jurisdiction and authority;
– Legal temporal parameters;
– Legal deontic operators;
– Qualifications;
– Semantic of negation;
– Behaviors.

The building blocks of Derivation RuleML are:

– **Predicates** are n-ary relations introduced by an `<atom>` element in RuleML. The main terms within an atom are variables `<var>` to be instantiated by values when the rules are applied, individual constants `<ind>`, data values `<data>`, and complex expressions `<expr>`. Derivation Rules are defined by an `<implies>` element which consists of a body part (`<body>`) with one or more atomicconditions connected via `<and>` or `<or>`, possibly negated by `<neg>` (for classical negation) or `<naf>` (for negation as failure), and of a conclusion part (`<head>`) that is implied by the body, where rule application can be in a forward or backward manner;

- **Facts** are stated as atoms deemed to be true, also introduced through the `<atom>` element;

- **Queries**, introduced with `<query>`, can be proven backward as top-down goals or forward via bottom-up processing, where several goals may be connected within a query, possibly negated.

Besides facts, derivation rules, and queries, RuleML defines further rule types such as integrity constraints and transformation rules. The language also allows including other elements and statements compliant with external ontologies. LegalRuleML will be thoroughly presented - and its implementation in the present research showed - in 4.5.

2.4.6.4. Reaction RuleML

Rules which trigger actions as a response to the detection of events have been extensively studied since the 1990s. Stemming from the early days of programming language where system events were used for interrupt and exception handling, active event-driven rules have received great attention in different areas such as active databases, real-time applications and system, network management tools and publish-subscribe systems [Paschke 2007].

Global reaction rules (see [Paschke et al. 2007]) typically follow the Event-Condition-Action (ECA) paradigm: "on *Event* and *Condition* do *Action*". These three terms are intended in [Dittrich et al. 1995] as having the following meaning:

- The **event** part specifies the signal that triggers the invocation of the rule;

- The **condition** part is a logical test that, if satisfied or evaluates to true, causes the action to be carried out;

- The **action** part consists of updates or invocations on the local data.

A global reaction rule is formalized as an extended ECA rule, represented in the KB as a 6-ary fact including the following elements: *time*, *event*, *condition*, *action*, *post condition*, *else*.

- The **time** part of an ECA rule defines a pre-condition (an explicitly stated temporal event defining a validity time or event processing window) which specifies a specific point or window of time at which the ECA rule should be processed by the ECA processor;

- **Event**, **condition** and **action** have the same role as in normal ECA rules;

- The **post-condition** is evaluated after the action has been executed. It might be used to prevent backtracking from different variable bindings via cuts or it might be used to apply post-conditional integrity and verification/validation tests in order to safeguard transactional knowledge updates in ECA rules;

– The **else** part defines an alternative action which is executed in case the ECA rule could not be completely executed.

Reaction RuleML [Paschke et al. 2006] encodes reaction rules as an XML-serialized sublanguage of RuleML: it adds various kinds of production, action, reaction, and KR temporal/event/action logic rules as well as (complex) event/action messages. The building blocks of Reaction RuleML are:

– One **general reaction rule form** that can be specialized by the selection of the constituent subparts, i.e. Condition-Action rules (production rules), Event-Action rules (trigger rules), Event-Condition-Action rules;

– Three **execution** styles (default value is *reasoning*):

  o *Active*: 'actively' polls/detects occurred events in global ECA style, e.g. by a ping on a service/system or a query on an internal or external event database;

  o *Messaging*: waits for incoming complex event message (inbound) and sends messages (outbound) as actions;

  o *Reasoning*: logical reasoning as e.g., in logic programming (derivation rules) and KR formalisms such as event/action/transition logics (as e.g. in Event Calculus, Situation Calculus, TAL formalizations);

– *Weak* and *strong* evaluation/execution **semantics interpretations** which are used to manage the "justification lifecycle" of local inner reaction rules in the derivation/execution process of the outer rules;

– An optional **object identifier**;

– An optional **metadata label** with i.e.. metadata authoring information;

– An optional set of **qualifications** (i.e. a validity value, fuzzy value, a defeasible priority value, etc.);

– A definition for inbound or outbound event **messages**.

A reaction rule might apply globally or be locally nested within other reaction or derivation rules. The most general syntax of a reaction RuleML[11] is the following:

```
<Rule style="active" evaluation="strong">

 <label> <!-- meta data --> </label>

 <scope> <!-- general scope of rule --> </scope>

 <qualification> <!-- qualifications --> </qualification>
```

[11] See Reaction RuleML, http://ruleml.org/reaction/.

```
 <oid> <!-- object identifier --> </oid>
 <on> <!-- event --> </on>
 <if> <!-- condition --> </if>
 <then> <!-- conclusion --> </then>
 <do> <!-- action --> </do>
 <after> <!-- postcondition --> </after>
 <else> <!-- else conclusion --> </else>
 <elseDo> <!-- else/alternative action --> </elseDo>
 <elseAfter> <!-- else postcondition --> </elseAfter>
</Rule>
```

2.4.7. The boundary between Ontology and Rules

LKIF rules are considerably more expressive than OWL, in particular with respect to the use of variables. In some cases users will inevitably resort to a rule even where a terminological axiom was intended. This makes it hard to maintain that there is any clear conceptual distinction, as there are independent technical reasons to make something a rule instead of a terminological statement.

LKIF has been designed to be sufficiently expressive to support the isomorphic modelling [Bench-Capon and Coenen, 1992] of legislation, at a very high level, in order to facilitate the development, validation and maintenance of knowledge bases by legal experts. The rule language is more expressive than formulas of first-order logic, let alone subsets of first-order logic, such as Horn clause logic or description logic, which have been developed due to interesting computational properties, such as (semi-)decidability or even tractability (see 1.2.2.4.). LKIF has instead been designed to optimize expressiveness for the legal domain, not computational efficiency. It is designed for use in interactive systems which help users to construct theories and arguments, as well as traditional expert systems, which interactively acquire facts and deduce propositions from these facts by applying rules. For argument construction tasks, the expressivity of the knowledge representation language is more important than the computational properties of the inference relation, since users are responsible for controlling the search for arguments. For traditional expert system, the computational properties of the inference relation may be more important, since the inference engine is expected to fully automatically derive logical conclusions from the facts input by the user. LKIF has been designed to be expressive enough for both kinds of systems, but when using LKIF for traditional expert systems, it is necessary to use a subset of LKIF which can be handled by inference engines with the desired computational properties.

The difference in expressiveness is unavoidable: LKIF needs more, and OWL cannot be much more expressive without losing its usefulness as a Semantic Web integrative technology. For OWL, real-time consistency checking is an important design issue. For the LKIF Rules, instead, tractability is much less of an issue, considering its semantics definition. In practical terms the implementer is free to draw the boundary between ontology and rules wherever he likes. It is possible to exclusively use rules, but the price to be paid is in its potential for reuse: alignment of ontologies is generally considered to be the first step to integration of knowledge bases, and by using rules one – firstly – signals that the information is considered to be domain-specific by its author, and secondly one loses subsumption as an organizing principle. On the other hand, putting obviously defeasible reasoning policies in the ontology will eventually cause inconsistencies, and will – considering OWL's role as a Semantic Web integrative technology – generally result in the ontology, and any rules dependent on it, not being used.

Under a certain perspective, this distinction represents different entrenchment degrees. Even if one takes the position that in the end everything is defeasible, one will usually want to impose some entrenchment ordering so that in the case of conflict some things give way more easily than others. This position is taken by [Boer, 2000], who notes the importance to legal knowledge engineering of a rhetorical hierarchy in legal argumentation, to the effect that one should prioritize technical arguments over normative arguments, normative arguments over epistemological arguments, and epistemological arguments over ontological arguments. Also outside the legal field, raising ontological issues ("Oh, that depends on what you mean with X") in an intellectual discussion that started out with a technical disagreement is generally interpreted as a sign of weakness of one's position. The rules versus ontology distinction can be used to model a number of phenomena that knowledge engineers encounter. In these cases they may want to impose such an entrenchment by prioritizing stronger forms of argument to weaker ones: "classical not" versus a negation as failure, not knowing whether something is the case versus knowing something is not the case, constitutive rules versus essential rules of an institution, and so on. Relations between these distinctions have been explored, and some have been shown to be special cases or alternative formulations of each other: for example, defeasible and constitutive rules can be represented as argumentation schemes (see [Governatori 2011]; negation as failure, unless, and assuming can be represented as autoepistemic expressions, as discussed in [Motik 2006]; essential rules can be represented as terminological axioms. For a proposed solution on this central issue on knowledge representation see section 4.4.2.

2.5. Judicial Reasoning in the Semantic Web

The development of the Semantic Web technologies in the last 10 years introduced brand new possibilities in terms of knowledge representation and automatic processing. As it has been already described, specific projects exploited these technologies to improve creation, workflow and management of different kinds of legal resources. Nevertheless, these technologies could not allow a complete support for legal reasoning, intended as a speculation on facts and rules that includes complex logic operations such as subsumption, analogy, distinction. Under this point of view, the Artificial Intelligence technologies introduced for the Semantic Web could do nothing more for lawyers than deduce the consequences from a precisely stated set of facts and legal rules. This makes many lawyers sceptical about the usefulness of such technologies, since this mechanical approach seems to leave out most of what is important in legal reasoning. To them, a case does not appear as a set of facts, but rather as a story told by a client, a story that has to be interpreted by the lawyer in order to identify the framework of applicable law. Depending on the interpretation and thus on the framework applied, several critical questions

can be applied to the story, in order to evaluate the importance of single factors and to qualify that story. In a legal claim, two incompatible stories concerning the same facts are examined in order to compare interpretations, factors and applicable law frameworks. This is true – even if to different extents – both for Civil Law and Common Law legal systems.
As described in 1.4.2., several projects tried to properly represent judicial reasoning with ITs, although outside of the Semantic Web. The main objective of the present thesis is to adapt the existing Semantic Web technologies to include legal argumentation, and to provide a set of requirement for a standard in the logic and proof layers of the Semantic Web layer cake.

Chapter 3

An Ontology Set Representing Judicial Decisions

"Data is a precious thing and will last longer than the systems themselves."

– Tim Berners-Lee.

"I see, these books are probably law books, and it is an essential part of the justice dispensed here that you should be condemned not only in innocence but also in ignorance."

- Franz Kafka, *The Trial*.

3.1. The Judicial Framework

The goal of the present research is to define a Semantic Web framework for precedent modelling, by using knowledge extracted from text, metadata, and rules [Bench-Capon 2009], while maintaining a strong text-to-knowledge morphism between legal text and legal concepts, in order to *fill the gap* between legal document and its semantics [Palmirani and Brighi 2010]. The input to the framework includes metadata associated with judicial concepts, and an ontology library representing the structure of case-law.

The research relies on the previous efforts of the community in the field of legal knowledge representation [Mommers 2010] and rule interchange for applications in the legal domain [Gordon et al. 2009]. The issue of implementing logics to represent judicial interpretation has already been faced in [Boella et al. 2010], albeit only for the purposes of a sample case. The aim of the present research is to apply these theories to a set of real legal documents, stressing the OWL axioms definitions as much as possible in order to enable them to provide a semantically powerful representation of the legal document and a solid ground for an argumentation system using a defeasible subset of predicate logics.

The ontology library is the cornerstone for a semantic tool that enriches the XML mark-up of precedents (the metadata), and supports legal reasoning. It appears that some new features of OWL2 unlock useful reasoning features for legal knowledge, especially if combined with defeasible rules. The main task is thus to formalize legal concepts and argumentation patterns contained in a judgement, with the following requirement: to check, validate and reuse the discourse of a judge - and the argumentation he produces - as expressed by judicial text.

In order to achieve this, four different models that make use of standard languages from the Semantic Web layer cake (Figure 3.1) have been used:

a. a **document metadata structure**, modelling the main parts of a judgement, and creating a bridge between a text and its semantic annotations of legal concepts;

b. a **legal core ontology**, modelling abstract legal concepts and institutions contained in a rule of law [Ceci and Palmirani 2012];

c. a **legal domain ontology**, modelling the main legal concepts in a specific domain concerned by case-law (e.g. contracts, e-commerce services, tort law, etc.);

d. an **argumentation system** [Ceci and Gordon 2012], modelling the structure of argumentation (arguments, counterarguments, premises, conclusions, rebuttals, proof standards, argument schemes, etc.).

The present chapter introduces the structure of the *core* and *domain ontologies* – points b. and c. - which have been designed to organize the metadata annotating the text of judicial decisions, and to infer relevant knowledge about precedents.

The metadata structure and the argumentation system are not described in this work: the metadata structure relies on the Akoma Ntoso standard (see 2.4.2.), while multiple solutions are being tested for building argumentation out of the ontology library: an application of the ontology library to the Carneades Argumentation System is described in Chapter 4, while future research could focus on SPINdle (see 6.2.) and a Drools application currently under development (see 6.3.2.).Section 3.2. presents the requirements and the methods for the design of the ontology library, and its features. Section 3.3. describes the application of the ontology library to judicial interpretation knowledge. In section 3.4., the method is exemplified with reference to a sample of Italian case law. Section 3.5. presents an evaluation of the ontology, section 3.6. discusses related work either in legal ontology or legal reasoning fields, and section 3.7. discusses some issues of the proposed solution.

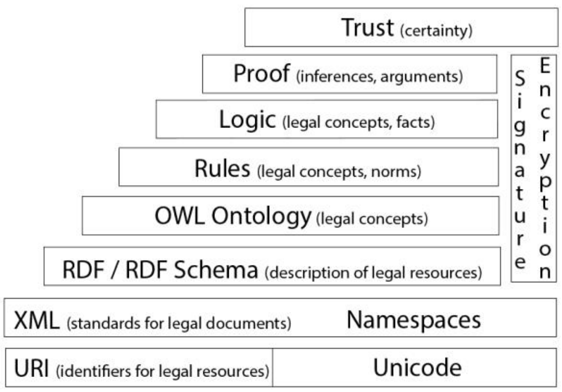


Fig. 3.1 – Tim Berners Lee's Semantic Web layer cake, adapted to the legal domain in [Sartor 2009].

3.1.1. Tasks and applications

The aim of the research described in the present chapter was to apply state-of-the-art techniques in ontology design and DL reasoning to knowledge from legal documents, stressing OWL2 axiomatization capabilities in order to provide an expressive representation of judicial documents, and a solid ground for an argumentation system using a defeasible subset of predicate logics.

Modelling judicial knowledge involves the representation of situations where strict deductive logic is not sufficient to reproduce the legal reasoning as performed by a judge. In particular, defeasible logics [Governatori and Rotolo 2004] seem needed to represent the legal rules underlying judicial reasoning. For example, many norms concerning contracts are not mandatory: they could be overruled by a different legal discipline through specific agreements between the parties. The problem of representing "defeasible" rules, in fact, is a core problem in legal knowledge representation.

On the other hand, argumentation theories (including the dialogue model of adjudication by [Prakken 2008], and argumentation schemes by [Gordon and Walton 2009]) introduce tools that are fundamental to perform effective reasoning on legal issues. This perspective adopts a procedural view on argumentation, which is necessary in order to properly represent those processes in an argument graph.

The purpose of the ontology presented in this article is however limited to enrich the metadata annotating a legal document by performing shallow reasoning on the knowledge base, and thus preparing it for additional reasoning performed by tools based on deontic defeasible logics and argumentation schemes. The ontology library has to satisfy the following functional requirements:

- **Text-to-knowledge morphism**: the aim is to design the knowledge that can be extracted from a a (textual) judicial decision, or a fragment of it, as a module in an ontology library, so that each module constitutes a particular morphism of the legal meaning expressed by that text [Palmirani et al. 2009];

- **Distinction between document layers**: the ontology must clearly distinguish between the medium and expression (the legal text), its meaning (the legal concepts and rules contained in the text), and the entities referred by the text. In principle, different (and even inconsistent) legal meanings can be expressed by a same legal text;

- **Shallow reasoning on judicial knowledge**: the ontology must enable reasoning on material circumstances, legal concepts and judicial interpretations contained in precedents, deriving inferences out of the legal concepts and elements involved in a judicial decision;

- **Querying**: being able to perform complex querying, e.g. by using SPARQL-DL [Sirin and Parsia 2007], on qualified parts of a judgement text. For example, performing queries that encode a question such as: "*retrieve all the judgements in the last year, with a dissenting opinion, in the e-commerce field, and where the main argument of the decision is the application of Consumer Law, art. 122*";

- **Supporting text summarization**: detecting relevant parts of a judicial text by reasoning on semantic annotations jointly with judicial ontologies;

- **Modularity**: the legal core ontology should define concepts common to as many domain ontologies as possible, which in turn should be automatically imported depending on the task at hand;

- **Supporting case-based reasoning**: performing legal case-based reasoning by using the ontology reasoner in combination with a set of rules, and a rule engine.

Judicial ontologies are intended to create an environment where the knowledge extracted from the decision text can be processed and managed, and reasoning on the judicial interpretation grounding the decision is made possible. Reasoning intends to satisfy the following domain requirements[12] (also known as *competency questions*, see [Gruninger and Fox 1994]):

- **Finding relevant precedents** that are not explicitly cited in the decision;

- **Validating the adjudications** of the judge on the claims brought forward by the parties during the trial on the basis of applicable rules, accepted evidence, and interpretation;

- **Suggesting legal rules/precedents/circumstances** that might lead to a different adjudication of the claim.

In order to reach those objectives, it is necessary to introduce particular structures (see section 3.2.2. and its subsections) to represent the instances of judicial interpretation in a special way. A judicial interpretation in fact:

- Performs an act of interpretation (taking into consideration a fact and applying a legal rule (legal status) to it;

- Is itself an interpretation of a legal text (being possible that the same combination of words gives rise to alternative interpretation acts, depending on the meaning given to the words).

The abstract categories of "qualifying expression" (see 3.2.2.1.) are aimed at capturing this layered stack of interpretations, while keeping an open approach (in order to maximize the results of the reasoning, since in the legal field even remote, apparently counterintuitive inferences may be decisive.

The structure of the ontology library also aims at an effective scaling from legal concepts to factors, up to dimensions and legal principles: all these concepts can be represented in the domain ontology, and the link to judicial concepts from the core ontology should foster semantic alignment between differently designed domain ontologies (the current ontology library alignments have not been tested yet). Eventually, practical applications of the ontology library include:

---

12 See Chapter 4 for an implementation of the ontology library into the Carneades Argumentation System.

- **Compliance check** of contract drafts, e.g. by using a plugin to a word processor that employs NLP techniques to recognize sentences and clauses that could be relevant under e.g. consumer law;
- **Juridical analysis tools** for legal professionals, enriching case-law collections by semantically relating and grouping precedents for lawyers to browse, making a precedent extraction process for legal cases easier and more effective;
- **Judgement management tools** for courts and tribunals, useful to evaluate and optimize judgements (e.g. integrated into a word processor to assist judges while writing judgements, so avoiding grounds for appeals due to missing elements in the decision's groundings);
- **Impact analysis tools** for legislators, providing a list of (common or uncommon) judicial interpretations for a given law, in order to take them into account when modifying that law;
- **Tools representing formalized legal doctrine and case law**, where legal experts could rely on a social platform to share their views and interpretations on a law or a precedent, e.g. by using a graphical interface and a formal argumentation structure instead of plain text.

3.2. Ontology Design

Judicial ontologies are designed in two modules (see also [Ceci and Palmirani 2012]):

- A **Core Ontology** describing the constituents of a precedent in terms of general concepts, through an extension to the LKIF-Core legal ontology (see 2.4.5.1.);
- a **Domain Ontology** representing the concepts and the rules expressed by the Italian *Codice del Consumo* (Consumer Code) and in *artt*. (articles) 1241 and 1242 of the Italian Civil Code, as well as all relevant knowledge extracted from a set of Italian judgements containing interpretation of private agreements in the light of those laws.

Our design method is based on a middle-out methodology: bottom-up for capturing and modelling the legal domain ontology, and top-down for modelling the core ontology classes and the argumentation theory components (see 1.3.). Middle-out methodology is implemented here by using pattern-based design [Gangemi 2007] with Ontology Design Patterns either extracted from judicial text or reused from the core ontology, and matched according to requirements.

The approach adopted is based on a multi-layer paradigm, where a legal resource is managed in separate levels that are linked to each other, and organized in order to allow multiple annotation, interpretation, and classification with representation redundancy. The syntactical approach is based on the following schema:

- **Text annotation in XML**: the Akoma Ntoso standard [Vitali 1997, Barabucci et al. 2010] grants proper mark-up of the structure of the judgement and of citations;

– **Metadata annotation**: the Akoma Ntoso *metadata* block captures not only the metadata concerning the lifecycle of the document (e.g. workflow of the trial, formal steps, jurisdiction, level of judgements), but also the legal qualification of relevant parts of the decision, such as the minority report or the dissenting opinion;

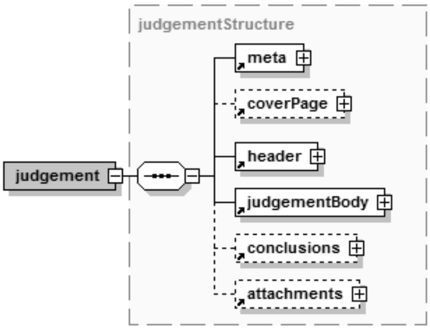


Fig. 3.2 – Judgment structure in Akoma Ntoso.

– **Ontology annotation**: external OWL definitions linked to the XML document are used;

– **Rules**: unfortunately OWL, even with the functionalities of version 2.0, is unable to represent complex and defeasible legal arguments. It is therefore necessary to extend the model with rule modelling for argumentation representation.

Evaluation has been performed on a sample set of Italian case law including 27 decisions of different grade (Tribunal, Court of Appeal, Cassation Court) concerning the legal field of oppressive clauses in Consumer Contracts. The matter is specifically disciplined in the Italian “Codice del Consumo” (Consumer Code), as well as in many non-Italian legal systems, so that an extension of this research to foreign decisions (and laws) can be envisaged.

Contract law is an interesting field because the (either automatic or manual) markup of contract parts allows the highlight of single clauses and their comparison to general rules as well as to case law concerning the matter. These possibilities can be used to introduce a semi-automatic compliance check of a contract draft.

The domain considered involves situations where strictly deductive logic is not sufficient to represent the legal reasoning as performed by a judge. In particular, defeasible logics [Governatori and Rotolo 2004] seem needed to represent the legal rules underlying judicial reasoning. For example, many norms concerning contracts are not mandatory: they could be overruled by a different legal discipline through specific agreements between the parties. The problem of representing "defeasible" rules, in fact, is a core problem in legal knowledge representation. Exploring how OWL2 could help designing the background for applying defeasible logic is therefore an important goal of the present research.

The software used to model the ontology (and from which the images of this paper are taken) is Protégé 4.1.0, supporting some of the features introduced by OWL2.

3.2.1. Judgement Structure

Judgement in Akoma Ntoso [Barabucci et al. 2010a] is a particular type of document modelled for detecting the main significant parts of a document about precedents (Figure 3.2): header for capturing the main information such as parties, court, neutral citation, document identification number; body for representing the main part of the judgement, including the decision; conclusion for detecting the signatures.

The body part is divided into four main blocks:

– **Introduction**, where usually (especially in common law de-cisions) the story of the trial is introduced;

– **Background**, dedicated to the description of the facts;

– **Motivation**, where the judge introduces the arguments supporting his decision;

– **Decision**, where the final outcome is given by the judge.

This division allows detecting facts and factors from the background: in the motivation arguments and counterarguments are detected, and in the decision lies the conclusion of the legal argumentation process. Those qualified fragments of text should be annotated by legal experts with the help of a special editor (e.g. Norma-Editor, presented in [Palmirani and Benigni 2007]) that is handy to create links between text, metadata and ontology classes.

3.2.2. Core Ontology

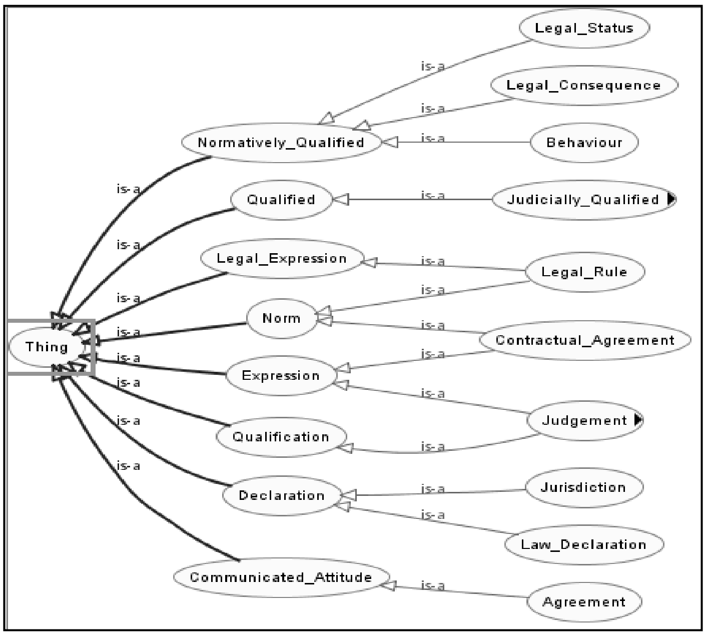


Fig. 3.3 – Core Ontology's specification of LKIF-Core. The central column defines categories already present in LKIF-Core, therefore their further classification is not contained in the graph.

The judicial core ontology[13] (Figure 3.3) introduces the main concepts in that legal domain, defining the classes that including entities extracted from judicial decisions. Core ontologies are domain-generic and not modeled upon a specific legal subject, however being the legal domain too large and heterogeneous, the model presented here has been conceived to represent interactions in Civil Law, especially as far as contracts, laws and judicial decisions are

---

13 http://codexml.cirsfid.unibo.it/ontologies/judging_contracts_core.owl.

concerned. For other domains, e.g. public contracts, administrative law, tort law, etc. adaptations are needed.

3.2.2.1. General Structure of the Core Ontology

The backbone of the Core Ontology is constituted by three LKIF-Core classes:

- `lkif:Qualificatory_Expression` (subclass of `Legal_Expression<lkif:Expression<lkif:Proposition<lkif:MentalObject`) represents a

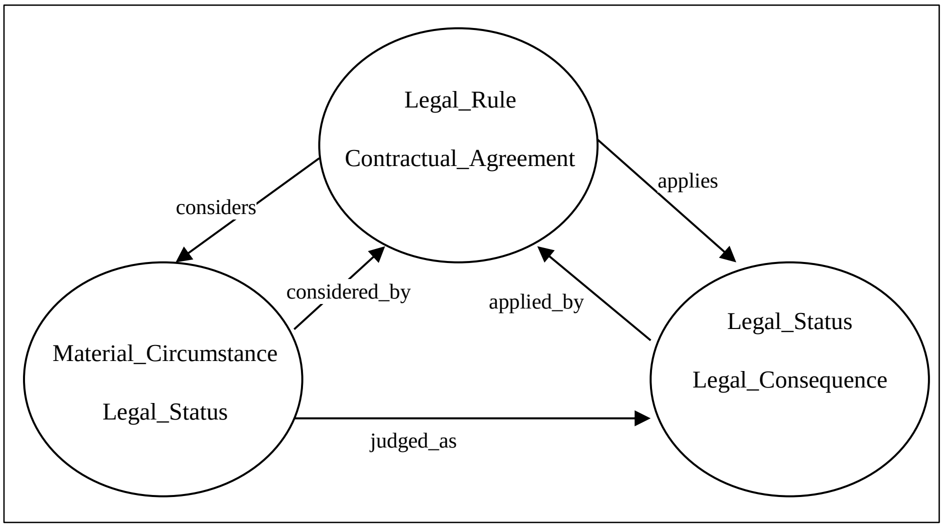


Fig. 3.4 – Interactions between qualifications, qualifiers, and qualified things.

  legal expression that ascribes a legal status to a person or an object. For example:

  - *x* is a citizen;
  - *x* is an intellectual work;
  - *x* is a technical invention.

- `lkif:Qualification` (SubClassOf `lkif:Proposition)` represents the legal act which contains the *qualificatory expression*. (i.e. contractual agreements, judgements).

- `lkif:Qualified` represents anything that is the object of some qualification.

On the basis of those classes the following ontology design pattern [Gangemi 2007] has been built. Since the main object to be represented in the present set of ontologies is the normative/judicial qualification brought forward by performative utterances (contractual agreements, legal rules and judicial interpretations), the classes presented above constitute the nucleus of the judicial core ontology. The lkif:`Qualification` and `lkif:Qualified` classes are linked only by a single property `(lkif:qualifies/lkif:qualified_by)`, but the need is rather to model an n-ary relation between three elements (Figure 3.4):

- A qualifying expression;

– A type of qualification;

– An object being qualified.

In order to represent this conceptualization, the property `qualifies` has been forked into two new properties: `considers` and `applies`. The first one, `considers` (modeled as a superclass of the LKIF-Core properties `evaluates`, `allows`, `disallows`) represents the object of the qualification. The second property, `applies`, shows *towards which concept* the qualification is made. The pattern applies to different entities as follows:

– A `Contractual_Agreement considers a Material_Circumstance and applies a Legal_Status;`

– A `Judicial_Interpretation considers` a `Material_Circumstance` and `applies` a `Legal_Status`;

– A `Legal_Rule considers` a `Legal_Status` and `applies` a `Legal_Consequence`;

– An `Adjudication considers` a `Judicial_Claim` and `applies` a `Judicial_Outcome`.

3.2.2.2. Qualifying Legal Expressions

To overcome the limited expressivity of the original LKIF-Core classes, a new class called

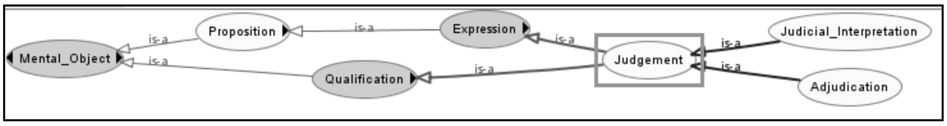


Fig. 3.5 – Visualization of the Judgement class as a Qualifying Legal Expression.

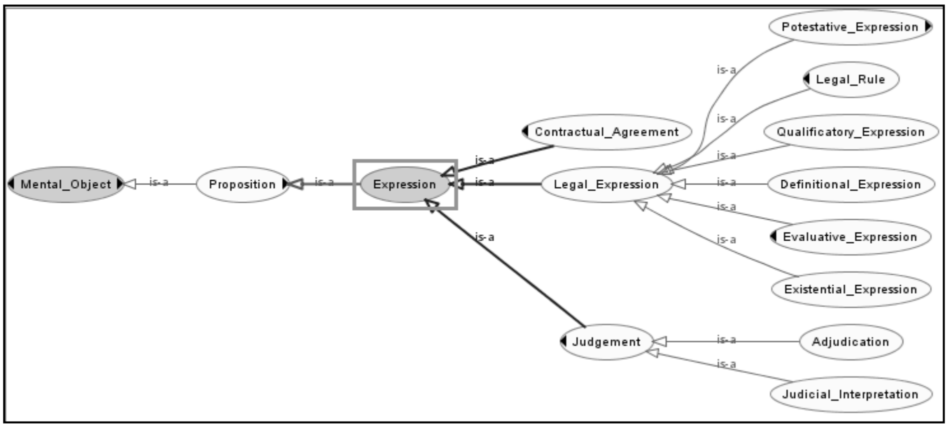


Fig. 3.6 – Visualization of the Expression class.

`Qualifying_Legal_Expression` (Figure 3.5) has been conceived, putting together the characteristics of the `Qualificatory_Expression` (Figure 3.6) and `Qualification` classes, enhanced by the specializations of the `qualifies` property. This class represents the formalization of *dispositions*, which in the sample case are the three legal expressions involved in contract law-related judicial decisions: Contractual_Agreement, Legal_Rule and Judgement.

As a subclass of Qualificatory_Expression qualifying legal expressions contain all the information related to their original "speech act": its semantics binds with the externalization, the legal power and the agents to ensure the representation of all aspects that may come into play when facing a legal issue (the legitimacy of the legislative body/court/legal party, the characteristics of the corresponding legal document, the identity/characteristics of

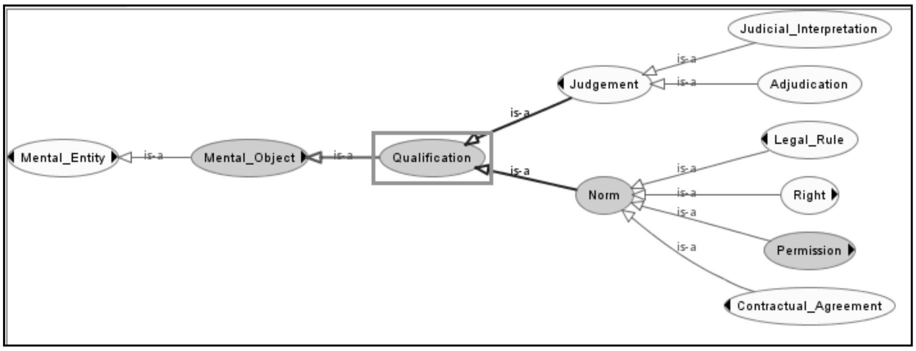


Fig. 3.7 – Visualization of the Qualification class.

people/bodies involved, etc.). Their main properties are "medium" and "attitude" (see below for a specification of the Medium, Attitude and Agent classes).As a subclass of Qualification (Figure 3.7), qualifying legal expressions contain the information related to the effects they have in the legal world: the legal categories / obligations / effects they create, modify or repeal.

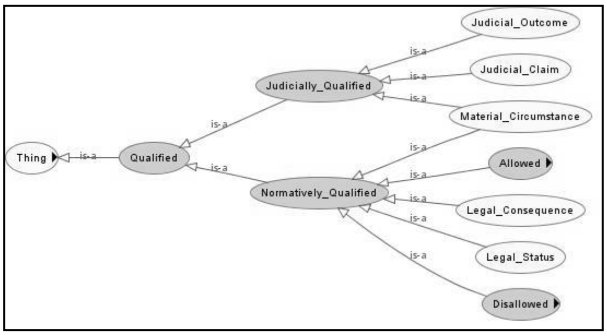


Fig. 3.8 - Visualization of the qualified class.

A subdivision can be made between one direct subclass (Judgement, which in this perspective is furtherly divided into the Judicial_Interpretation and Adjudication subclasses) and two subclasses of Norm (Legal_Rule and Contractual_Agreement).

As explained before, the property qualifies - linking the qualifying expression to the Qualified expression - has been forked into two new properties: considers and applies, representing respectively the direct object and the "destination" of the qualification.

3.2.2.3. Qualified Expressions

All the ranges of the considers and applies properties presented above are subclasses of the Qualified class (Figure 3.8). Its subclasses are Normatively_Qualified, a class already present in LKIF-Core, and Judicially_Qualified, created anew.

Normatively_Qualified expressions include Material_Circumstance, Legal_Status and Legal_Consequence. They represent the expressions that can be directly bound to a Norm: while Material_Circumstance represents any fact or act which is taken into consideration by the Norm, Legal_Status represents an institutional fact (i.e. fulfillment of contract, oppressive clause, contract breach) that is normally considered_by a Legal_Rule and applied_by a Contractual_Agreement or a Judgement. Please note that the link between a Contractual_Agreement and the Legal_Status it applies is a "weak" link until a Judicial_Interpretation has confirmed (or denied) it. Finally, Legal_Consequence represents the sanction provided by the law in the presence of some Legal_Status or Material_Circumstance. It covers all cases when the Legal_Rule considers some Normatively_Qualified expression, but does not simply allows, disallows or evaluates it.

Judicially_Qualified expressions include Judicial_Claim, Judicial_Outcome and all elements taken into consideration during a legal proceeding (i.e. Contractual_Agreeement, but also Legal_Rule, expecially in Cassation Court and Costitutional Court sentences). Judicial_Claim is the claim of the legal proceeding. It is considered_by an Adjudication, the answer of the judge to the claim (subclass of Qualification>Judgement). The content of the answer (rebuttal/acceptation of the claim or any other possible outcome foreseen by the law) is represented by the Judicial_Outcome class, applied_by the Adjudication. So the representation is the following: a Judicial_Claim is considered_by an Adjudication that applies a Judicial_Outcome.

3.2.2.4. The judged_as Property Chain

The miscellaneous elements that can be taken into consideration during a legal proceeding are included in the Judicially_Qualified class as long as they are actually considered_by some Judicial_Interpretation. So, for example, a Contractual_Agreement can be considered_by some Judicial_Interpretation who applies some Legal_Status to it (i.e. the agreement is oppressive, is inefficacious, represents an arbitration clause, is specifically signed by both parties). In these cases, a OWL2 property chain (see 3.2.3.1. for a description of this feature and of its usage in the present research) directly links the Contractual_Agreement to the Legal_Status judicially applied to it. This *strong* link, represented by the property "judged_as", is the the fundamental information to be represented – and managed – through this set of ontologies. This particular model of qualification can be formalized and reused through a corresponding ontology design pattern [Gangemi 2007].

3.2.2.5. Mediums, Propositional Attitudes and Agents

These LKIF-Core classes describe the background of an Expression. The Medium class identifies the support through which the proposition is expressed. It does not represent the material support of the Expression instance but rather its *genus* (Contract, Precedent, Code).

The Propositional_Attitude class was specified with the Jurisdiction, Law_Declaration and Agreement subclasses, representing the enabling powers that stand behind a Judgement, a Legal_Rule and a Contractual_Agreement, respectively. On the contrary, to represent the authors of a Qualifying Legal Expression there was no need to specify the subclasses of Agent already present in LKIF-Core (Person and Organization). This knowledge about agents and attitudes can be important in some judicial cases: i.e. if a claim is based on the lack of contractual power by one of the parties, or on the identity/characteristics of a part, or on the lack of force by some law or other regulation (which can in turn depend by the lack of legitimacy of one of its authors). Also the modelling of roles (already present in LKIF) can be very useful in representing critical factors of particular precedents.

3.2.2.6. Modularity of the Core Ontology

The expansion brought by the Core Ontology to the LKIF-Core concepts is currently oriented to the representation of the elements involved in civil-law cases regarding contract law.

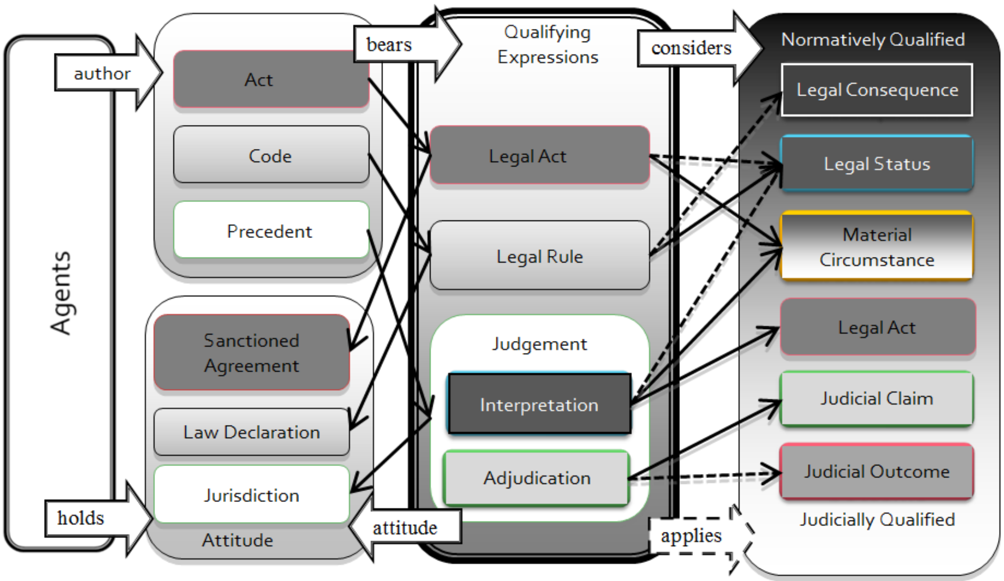


Fig. 3.9 – The Core Ontology graph. Boxes represent classes. Continuous arrows represent either the bears, attitude or considers properties. Dashed lines represent the applies property.

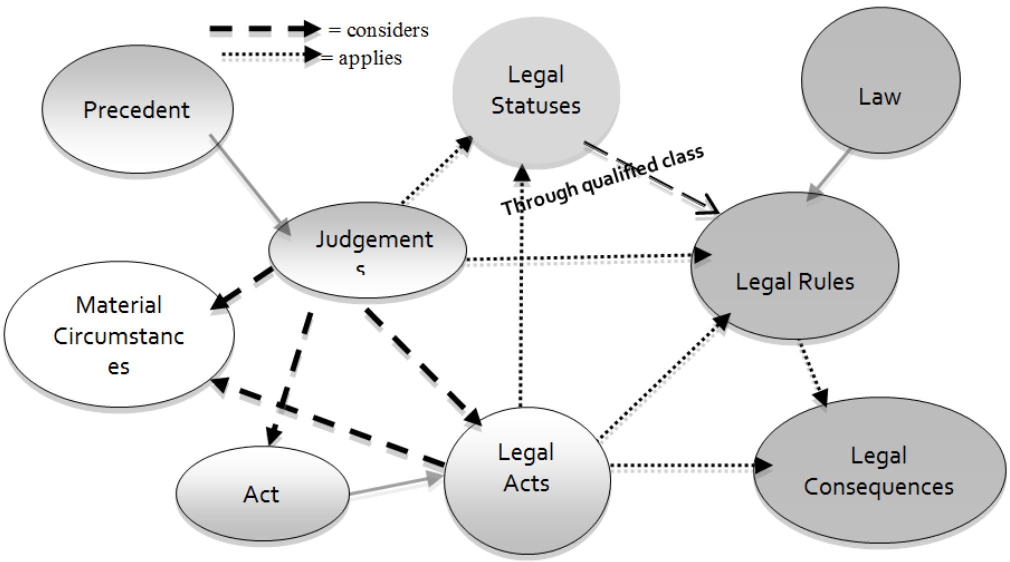


Fig. 3.10 – semantic relations between represented knowledge. The dashed line "Through qualified class" means that the connection from legal statuses to legal rules is ensured through a qualified class (see 3.2.2.).

Nevertheless, the Core Ontology provides general – and relatively open - categories for this kind of judicial activity to be represented, and can therefore be considered as a core to be "expanded" with categorization from other branches of law, but not to be "substituted", since the basic concepts introduced here may come into play also in judgements concerning different subjects.

Figure 3.9 represents the classes and properties of the core ontology. Figure 3.10 shows the same information, but allows to better understand the connection between the classes of the ontology.

3.2.3. Domain Ontology

Following this structure, the metadata taken from judicial documents are represented in the Domain Ontology[14]. The modelling was carried out manually by an expert in the legal subject, which actually represents the only viable choice in the legal domain, albeit giving rise to important bottleneck issues (see below 3.6.1.). Also, building a legal domain ontology is similar to writing a piece of legal doctrine, thus it should be manually achieved in such a way as to maintain a reference to the author of the model, following an open approach (i.e. allowing different modelling of the same concept by different authors).

3.2.3.1. Modelling of laws

Fig. 3.11 - Stated property assertion of a Legal Rule instance.

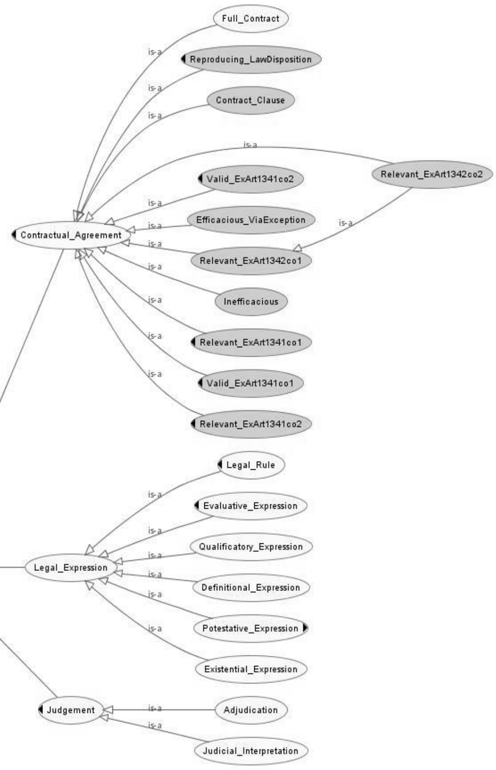


Fig. 3.12 – Visualization of the expression class, highlighting the subclasses of Contractual_Agreement introduced by the legal rules.

[14] http://codexml.cirsfid.unibo.it/ontologies/judging_contracts_domain.owl.

The laws involved in the domain are represented into the ontology in a quite complex fashion, in order to allow full expressivity of their deontic powers. First of all, they are represented as instances of the `Legal_Rule` class, whose only stated property is to apply the `Legal_Consequence` indicated in the head of the legal rule (Figure 3.11). The reasoner will infer knowledge about the rule, linking it (through the `considers` property) to the contractual agreements which fall under the scope of that norm.

Legal rules are also represented through anonymous subclasses of the `Normatively_Qualified` class (figure 3.12), called `Relevant_Ex<rulename>` (*ex* is the latin proposition for indicating a source). An axiom stating the requirements for an instance to be relevant under the legal rule is included in the description of the class, as well as an equivalence linking each of its instances to the legal rule, through the property `considered_by` (figure 3.13). Please notice that these anonymous classes are classified under the `Contractual_Agreement` class: that is, because the effect of the legal rule in this context is to enrich the definition of `Contractual_Agreement`, adding subdivisions which depend on the legal framework created by the legal rules of the domain.

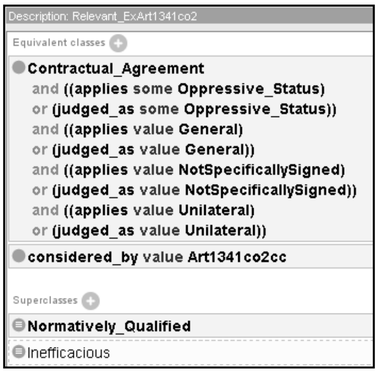


Fig. 3.13 - Axiom for classification of Contractual Agreements under the legal rule Art. 1341 comma 2.

3.2.3.2. Modelling of contracts

A contract is a composition of one or more `Contractual_Agreement`s (a `Contract` for the whole, multiple `Contract_Clause`s for its parts, an example being provided in figure 3.14), each of which represents an obligation arising from the contract. All components of the contract share the same `Attitude` (the "meeting of minds" between the `Agent`s) and `Medium` (the kind of support in which the expression is contained. A `Contractual_Agreement` normally `considers` some `Material_Circumstance` and `applies` some `Legal_Status` to it.

In the actual model, the material circumstances considered by the contractual agreement were not included: that is, because this has no relevance when capturing the sheer interpretation instances these agreement undergo: it would rather become useful when delving deeper into the single interpretation, capturing the smaller factors which led to that specific interpretation.

3.2.3.3. Modelling of judicial decisions

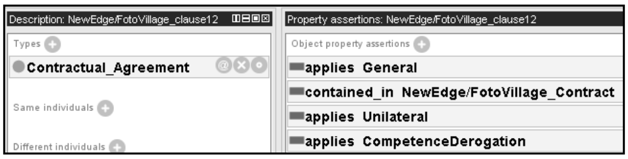


Fig. 3.14 - Description and property assertions of the contract clause's content.

The Judgement class includes an instance identifying the case as a whole (the precedent) and several ones identifying its parts: at least an Adjudication and one or more Judicial_Interpretations (figure 3.15). They share a common attitude (a Jurisdiction power) a Precedent medium and some agents (claimant, defendant, and court). The Adjudication contains the Judicial_Outcome of the Judicial_Claim. (it considers the claim and applies the outcome), while the Judicial_Interpretation considers a Material_Circumstance and applies one or more Legal_Status (and zero or more Precedents) to it. The precedents cited by the judge in the decision are added directly to the Interpretation instance: the reasoner is then capable of distinguishing between legal statuses and precedents, the latter being searchable in queries and other information retrieval applications. Rules expressed by precedents (i.e., if a clause is signed through a recall at the end of the document, it is specifically signed) can be modeled in the same way as legal rules are.

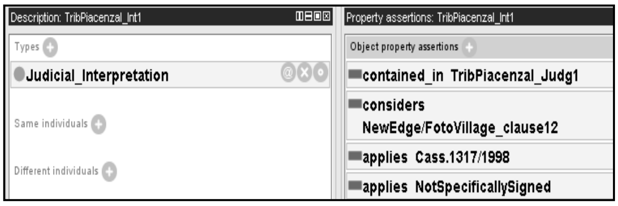


Fig. 3.15 - Description and property assertions of the judicial interpretation.

3.2.3.4. Reasoning on the knowledge base

The consistency of the Knowledge Base was checked with the Hermit 1.3.6[15] reasoner. This tool was built to extract data from the OWL ontology, but could also be used to check if the ontology gives a unique and correct answer to some formalized question (i.e. asking about the validity of some proof, or about the qualification of factual events under legal principles). When a Contractual_Agreement (the expression brought by a Contract_Clause) is considered_by some Judicial_Interpretation, the ontology gathers all relevant information on the documents involved: contract parties, judicial actors, legal status applied to the agreement (eventually in comparison to the one suggested by the contract/judicial parties), the law rules which are relevant to the legal status, the final adjudication of the claim, the part played in it by the interpreted agreement, and so on.

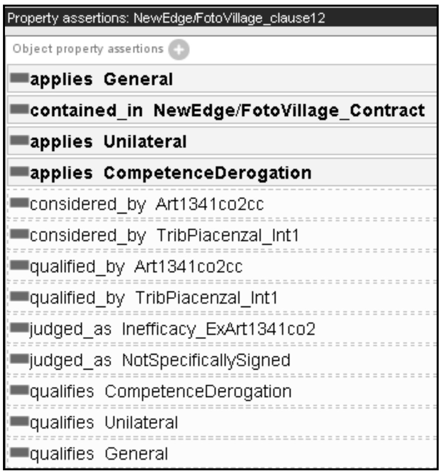


Fig. 3.16 - Inferred knowledge on the Contractual Agreement instance.

The first objective for gathering all this semantically-rich information is advanced querying on precedents, but more can be achieved by combining different Judicial_Interpretations with knowledge coming from the contract and the applicable law: the ontology reasoner is in fact capable of predicting – to some extents – the

15 http://hermit-reasoner.com/.

outcome of the judge (i.e. predicting that a clause will be judged as valid/invalid) and to run inferences about the agreement (for example, as interpreted, the clause in the example of figure 3.16 is relevant for the legal rule contained in article 1342 comma 2 of Italian Civil Code, and inefficacious in the light of the same norm).

This inferred knowledge is important for two reasons: *a.* by "predicting" the judge's final statement on the clause (even if not that on the claim), this knowledge represents a logic and deontic check on the legal consequences the judge takes from its interpretation; *b.* it gives a fundamental element for the argumentation system to support the explanation of the adjudication of the claim. The argumentation system, in fact, will be able to use the (stated and inferred) elements of the decision's groundings to support and explain the `Adjudication` contained in the last part of the judgement.

3.2.4. OWL2 Constructs Used

OWL2 (see [W3C Consortium 2012] and 2.2.4.) is one of the latest standard for the Semantic Web, and is relevant to any project willing to contribute to the huge network of data that is being built on the Web. An objective of the present research is to explore how OWL2 could help designing the background for the application of defeasible logic. OWL in general is not designed for managing defeasibility directly, being only able to capture the static factual and legal knowledge to be reused in the rule layer. Nevertheless, the gap between ontology and rules is often underestimated, and the benefits coming from OWL2 have not yet been considered in detail. For this reason, well aware of the limitations of OWL2 in representing defeasible logics, one aim of the present research is to investigate how far OWL2 can be used in order to improve performance, computability, and management of classes in a defeasible logic context.

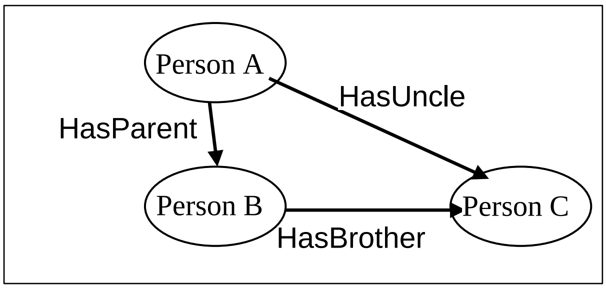


Fig. 3.17 – An example of property chain.

OWL2 introduces several features to the original Web Ontology Language, some of which allow a richer representation of knowledge, especially when dealing with properties and datatypes. Some of these would be useful, but also lead to a great increase of complexity in the models: for example, in order to exploit disjointness between properties it would be necessary to create as many properties as possible statuses, which in turn would greatly affect computability. On the contrary, some of these new constructs concerning properties deserve attention because they could enhance expressivity without affecting (or even reducing) the complexity of the model built so far. OWL functional syntax will be used in examples throughout the paper.

3.2.4.1. Property Chains

The OWL2 construct `ObjectPropertyChain` used within a `SubObjectPropertyOf` axiom allows a property to be defined as the composition of several properties as in Figure 3.17. Such axioms are known as *complex role inclusions* in SROIQ. The present research relies on a particular property chain useful in the judicial domain. The property chain:

```
considered_by o applies SubObjectPropertyOf judged_as
```

is represented in Figure 3.18, and is used in two different ways (in interpretations, as in the figure, and in rule applications) to create a direct interpretational link between a *material circumstance* and a *legal status*.

When a `Judicial_Interpretation` `considers` a `Material_Circumstance` and `applies` a `Legal_Status`, the `judged_as` property chain comes into play and creates a direct link between the circumstance and its status, that link being distinguished from the indirect one introduced by the contract (represented by the property applies). Reasoners will therefore treat these two links accordingly.

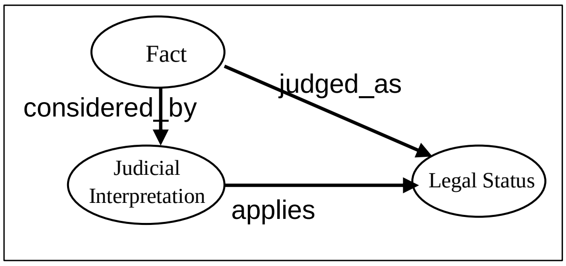


Fig. 3.18 – The property chain judged_as.

On the other hand, the legal rule axiom works through an "anonymous qualified class" which links all relevant expressions to the legal rule instance through the `considered_by` property, and the legal rule applies a legal consequence. The `judged_as` property chain unifies the two properties (from the qualified expression to the law, and from the law to the legal consequence) and brings their semantics to the surface by creating a direct property linking the contract clause to its status (`judged_as Inefficacy`).

A better use of the OWL2 property chains could lead to an ever more direct and complete solution, mainly by removing the need for the anonymous subclass in order to identify the clause instances `considered_by` the relevant law. In the actual version of the ontology, in fact, the property chain `judged_as` connects a material instance (i.e. contract clause) to a legal status or legal consequence (i.e. `oppressive`, `inefficacious`) via a judicial interpretation. With the open world approach, this creates a sprawling of `judged_as` chains being applied to the metadata. All of these inferences are correct; nevertheless, they greatly increase the number of triples in the ontology. In order for the ontology to manage a big knowledge base and to perform deep reasoning on it, it is therefore necessary to prune chain-based inferences in order to retain only those that are interesting for the task at hand. Since pruning would eliminate semantic content actually existing in legal documents, it has to be performed depending on the task of the rules application.

3.2.4.2. Negative object properties

A negative object property assertion such as:

```
NegativeObjectPropertyAssertion(OP a b)
```

states that the individual `a1` is not connected by the object property `OP` to the individual `a2`. E.g. given an ontology including the following axiom:

```
NegativeObjectPropertyAssertion(hasSon Peter Meg )
```

the ontology becomes inconsistent if it is extended with the following assertion:

```
ObjectPropertyAssertion(hasSon Peter Meg)
```

Negative object property assertions are useful to avoid complicated workarounds for negating assertions. For example, the legal status `NotSpecificallySigned` and more constructs are needed in OWL1 in order to represent the statement that a certain status is not SpecificallySigned, e.g.:

```
EquivalentClasses(SpecificallySigned?
ObjectOneOf(NotSpecificallySigned SpecificallySigned))

DifferentIndividuals(SpecificallySigned NotSpecificallySigned)

ObjectPropertyAssertion(applies ContractA NotSpecificallySigned)
```

but in OWL2 the following construct is sufficient:

```
NegativeObjectPropertyAssertion(applies                  ContractA
SpecificallySigned)
```

3.2.4.3. Keys

A `HasKey` axiom states that each *named* instance of a class is uniquely identified by a set of data or object properties assertions - that is, if two named instances of the class coincide on values for each of key properties, then those two individuals are the same. This feature is useful for identifying the unique elements in a judicial claim, e.g. the parties, the contract, the norm, and the decision itself.

3.2.4.4. Annotation properties

OWL1 allows extra-logical annotations to be added to ontology entities, but does not allow annotation of axioms. OWL2 allows annotations on ontologies, entities, anonymous individuals, axioms, and annotations themselves.

This feature is used in the judicial ontology library to provide a full-fledged information structure about the author of each piece of the model (i.e., who modeled a certain axiom, which legal text it refers to, and who/when/how was the original legal text created). Moreover, it is possible to give domains (AnnotationPropertyDomain) and ranges (AnnotationPropertyRange) to annotation properties, as well as organize them in hierarchies (SubAnnotationPropertyOf). These special axioms have no formal meaning in OWL2 direct semantics, but carry the standard RDF semantics in RDF-based semantics, via the mapping to RDF vocabulary.

3.2.4.5. N-ary datatypes

In OWL it is not possible to represent relationships between values for one object, e.g., to represent that a square is a rectangle whose length equals its width. N-ary datatype support was not added to OWL2 because it was unclear what support should be added. However, OWL2 includes all syntactic constructs needed for implementing n-ary datatypes. The Data Range Extension: Linear Equations note proposes an extension to OWL2 for defining data ranges in terms of linear (in)equations with rational coefficients. This kind of equations is of high importance in the process of identifying individuals to classify under a legal ontology framework, on the basis of a quantitative evaluation of the relationship between several factors.

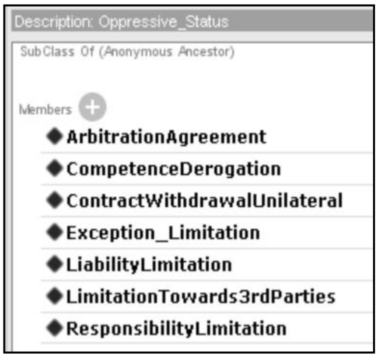


Fig. 3.19 - The list of legal statuses classified as oppressive.

3.2.4.6. Property qualified cardinality restrictions

While OWL1 allows for restrictions on the number of instances of a property (i.e. for defining persons that have at least three children) it does not provide means to constrain object or data cardinality (qualified cardinality restrictions, i.e. for specifying the class of persons that have at least three children who are girls). In OWL2 both qualified and unqualified cardinality restrictions are possible through the constructs: ObjectMinCardinality, ObjectMaxCardinality, and ObjectExactCardinality (respectively DataMinCardinality, DataMaxCardinality, and DataExactCardinality). These restrictions, together with n-ary datatypes, are fundamental to enrich the ontology with elements ensuring automatic classifications of qualified properties (e.g the minimum income needed for a claim to be classified under a certain category).

3.3. An Example of judgement Modelling

The modelling of the ontology is explained here through a simple example of data insertion and knowledge management by the Domain Ontology. Following is a description of the case to be modeled:

In the decision given by the 1st section of the Court of Piacenza on July 9th, 2009[16], concerning contractual obligations between two small enterprises ("New Edge sas" and "Fotovillage srl", from now on α and β), the judge had to decide whether clause 12 of α/β contract, concerning the competent judge (Milan instead of Piacenza) could be applied. The judge cites art. 1341 comma 2 of Italian Civil Code which says: "a general and unilateral clause concerning competence derogation is invalid unless specifically signed". In the contract signed by the parties there is a distinct box for a "specific signing" where all the clauses of the contract are recalled (by their number). The judge, with the support of precedents (he cites 9 Cassation Court sentences) interprets the "specific signing" as not being fulfilled through a generic recall of all the clauses, and therefore declares clause 12 of α/β contract invalid and inefficacious. The claim of inefficacy of clause 12, brought forward by α, is thus accepted, undercutting the claim of a lack of competence by the judge of Piacenza, brought forward by β, which is rejected.

In order to represent the knowledge contained in that judgement text, three documents have to be modelled: Art. 1341 comma 2 of Italian Civil Code, the contract between the two enterprises α and β, and the decision by the Court of Piacenza.

3.3.1. Modelling of the law

[16] Sent. N. 507 del 9 Luglio 2009, Tribunale di Piacenza, giudice dott. Morlini.

Following is the law disposition involved in the judicial decision:

Article 1341 comma 2 of Italian Civil Code: Clauses concerning arbitration, competence derogation, unilateral contract withdrawal, and limitations to: exceptions, liability, responsibility, and towards third parties, are inefficacious unless they are specifically signed by writing.

The disposition is represented as a Qualifying Legal Expression (`Legal_Rule`) called “art1341Co2” (with a `Code` medium, a `Law_Declaration` attitude and a `Parliament` as agent) and the qualified class `Relevant_ExArt1341co2`. As seen in 3.2.3.1., a `Legal_Rule considers` a (combination of) `Legal_Status`(es) and `applies` a `Legal_Consequence` (or a deontic operator). Therefore any individual which has the characteristics required by the law is `considered_by` the `Legal_Rule`, which in turn `allows/disallows/evaluates` or `applies` some `Legal_Consequence` to it. In the example of figure 15, each `Contractual_Agreement` which `applies` “General”, “Unilateral”, “NotSpecificallySigned” and an `Oppressive_Status` (Figure 3.19) will be `considered_by` *art1341Co2*, which in turn `applies` the `Legal_Consequence` of “invalidityExArt1341co2”. The individuals *competentJudge* and *notSpecificallySigned* are thus created as `Legal_Status`es that can be `considered_by` a `Legal_Rule` and `applied_by` a `Contractual_Agreement`, and the individual “invalidityExArt1341co2” is created as a `Legal_Consequence` `applied_by` the `Legal_Rule` “art1341Co2”.

Fig. 3.20 - Stated property assertions for the sample agreement.

3.3.2. Modelling of the contract clause

The `Contract_Clause` “α/βClause12” (Figure 3.20) is created and linked to a `Contractual_Agreement` which applies the `Legal_Statuses` of “General”, “Unilateral” and “CompetenceDerogation”. This is done because there is no argue between the parties about whether clause 12 concerns a competence derogation. However, as explained before, this kind of link is a “weak” one, considering that the contractual parties have no power to force a legal status into a contract, and that reconducting a contractual agreement to the legal figure it evokes is the main activity brought forward by judicial interpretation in the contracts field. For this reason, the property “applies” related to a `Legal_Status` is weak when its domain is a `Contractual_Agreement`, and prone to be overridden by a contrasting application performed by a `Judicial_Interpretation`.

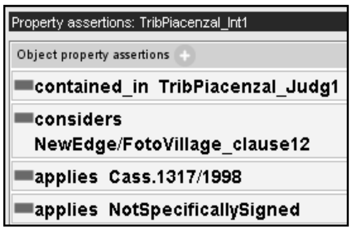


Fig. 3.21 - Stated property assertions of the sample judicial interpretation.

3.3.3. Modelling of the judicial interpretation

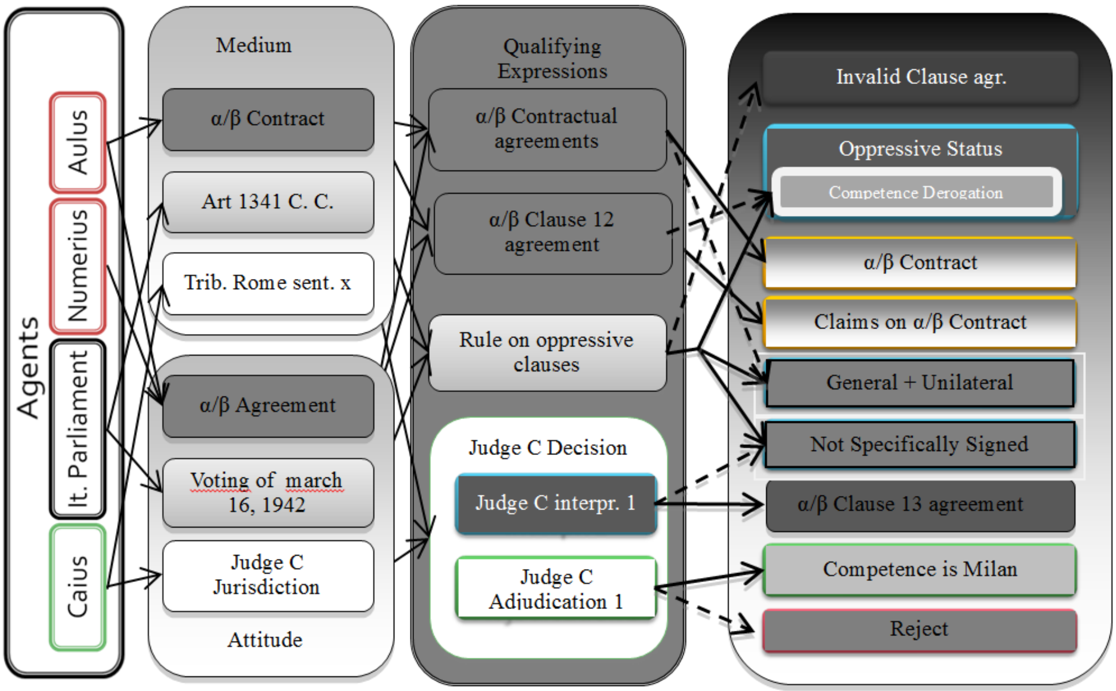


Fig. 3.22 - The graph showing the model of the sample case. The general classes of fig. 11 have been substituted with the sample instances. The properties (arrows) connect the same classes of the core ontology.

The Judgement instance is created, as well as its components (single interpretation instances, adjudication...). Among them, the *tribPiacenzaI_Int1* `Judicial_Interpretation` is created (Figure 3.21): it considers the `Contractual_Agreement` contained in *α/βClause12* and `applies` the *notSpecificallySigned* `Legal_Status`. The instance contains also a reference to the precedent (Cass.1317/1998), which represent a semantically-searchable information on the interpretation instance. Figure 3.22 shows all the elements created for the various classes, and the relations between them.

3.3.4. Reasoning on the knowledge base

In the example, when all the relevant knowledge is represented into the ontology, the reasoner is capable of inferring that "The agreement contained in clause 12 of the α/β contract is invalid ex article 1341 comma 2"(Figure 3.23). As already explained, this result is reached through a subclass of the `Contractual_Agreement` and `Qualified` classes, defined by an axiom representing the rule of law. Clauses that fulfill the axiom are automatically classified in that class, and thus `considered_by` the proper law. At this point, a property chain links the clause to the legal consequence through the legal rule (clause is

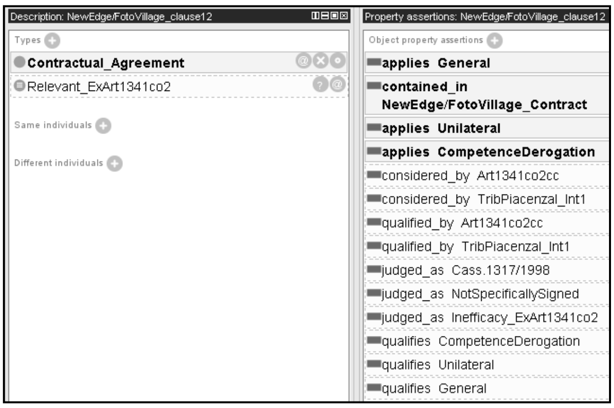


Fig. 3.23 - Inferred Description and property assertions of the contract clause's content.

considered_by the law which applies a legal consequence , thus the clause is judged_as th legal rule). The judged_as property thus gives the clause its final (efficacy/inefficacy) status under that law. Figure 3.24 explains the whole process as a list of axioms verified by the ontology reasoner.

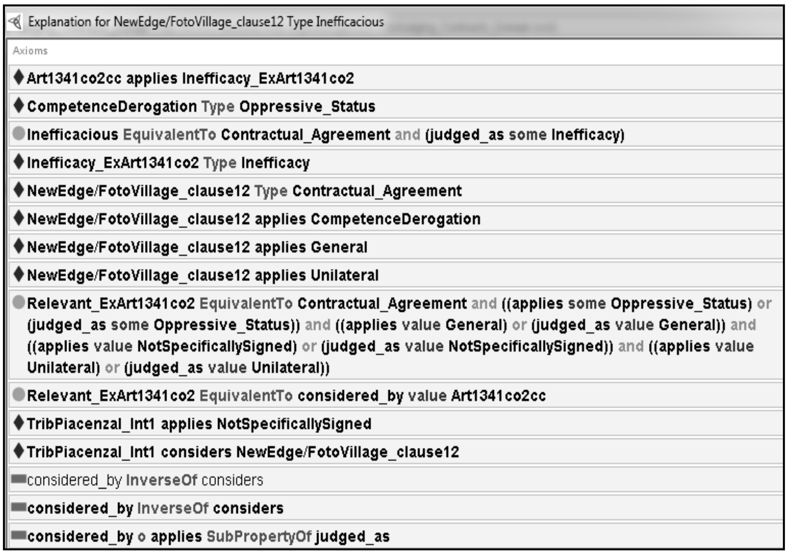


Fig. 3.24 - Explanation for the sample agreement being inefficacious.

3.4. Evaluation of the ontology library

The ontology library, in its sample taken from real judicial decisions, proved to meet the requirements of:

- **Text-to-knowledge morphism**: the ontology can correctly classify all instances representing fragments of text. The connection to the Akoma Ntoso markup language ensures the identification and management of those fragments of text and of the legal concepts they contain.

- **Distinction between document layers**: The qualifying expression class constitutes the main expressive element, introducing an n-ary relation that ignites the reasoning engine. Its instances can refer to the same text fragment, yet represent different (and potentially inconsistent) interpretations of that text. Moreover, the LKIF-Core's Medium class allows to represent different manifestations of the same expression;

- **Shallow reasoning on judgement's semantics**: the Domain Ontology can perform reasoning on a material circumstance’s relevancy under a certain law. The property chain judged_as and the axioms for law relevancy and legal consequence application allow the reasoner to complete the framework, also with the purpose of easening the effort needed to model all knowledge contained in the ontology. These axioms could also be used to support tools that automatically complete partially-modeled documents;

- **Querying**: the considers/applies properties allow complex querying on the knowledge base, and the judged_as shortcuts provide semantic sugar in this perspective. Querying on temporal parameters is not yet possible due to limits in LKIF-Core language:

solutions for this are being achieved through emerging standards for rules such as LegalRuleML.

- **Modularity***:* the layered (core/domain) structure of the ontology library renders domain ontologies independent between each other - and yet consistent, through their compliance to the core ontology template.

- **Supporting text summarization**: the ontology library supports the identification of dispositions and decision's groundings inside a judicial decision.

- **Supporting case-based reasoning**: An argumentation system has been built on a *lite* version of the ontology library. The axioms concerning law relevancy and law application have been removed from the ontology and moved to the rules layer, in order to have them applied not only on the ontology library's knowledge base, but also on the new knowledge derived from the application of the rules. Results of this can be found in [Ceci and Gordon 2012].

Computability was not an issue in the last ontology library version (<5 seconds reasoning time on a Intel i5@3.30 Ghz), while the Carneades reasoner was moderately encumbered by the application of the rules to the ontology (8-15 seconds in the example described in Chapter 4). This could be improved by optimizing the reasoner and/or with a further refinement of the ontology (and rules) structure.

3.4.1. Related Work

The framework presented in this paper relies on the previous efforts of the community in the field of legal knowledge representation [Boer et al. 2008] and rule interchange for applications in the legal domain [Gordon et al. 2009]. The issue of implementing logics to represent judicial interpretation has already been faced in [Boella et al. 2010], albeit only for the purposes of a sample case.

The methods applied for the construction of the core legal ontology are similar to those used for [Casanovas et al. 2005], an online repository of legal knowledge to provide answers to issues related to legal procedures. The main difference between the two approaches is that the latter relies on application of NLP techniques to user-generated questions in order to return the correct answer. The judicial ontology, instead, extracts information from official legal documents (laws, decisions, legal doctrine), whose content classification requires the intervention of a legal expert. Furthermore, the ontology in [Casanovas et al. 2005] focuses on legal procedure, while the present ontology concerns mainly the legal operations carried out by the judge in a decision, mainly judicial interpretations seen as subsumption of material facts or circumstances under abstract legal categories.

The project presented in [Savvas and Bassiliades 2009] focuses on a lower layer of the Semantic Web, concerning document structure and data interchange between different legal documents. For the same purposes, the present project relies on Akoma Ntoso (see 2.4.2.). Besides its being foucused on administrative procedures, the project in [Savvas and Bassiliades 2009] shows a rather interesting view on the procedural aspects of legal phenomena, which is something this ontology does not achieve, being this task demanded to an argumentation layer placed on top of the ontology layer.

[Despres and Szulman 2004] shows an automatic construction of an ontology concerning the language of a legislative text. The project is focused on the linguistic aspects, in particular on the use of NLP techniques to normalize and formalize the text in a set of concepts organized in an ontology. The ontology is built around DOLCE and LRI-Core, which makes it likely to be aligned with the ontology presented in this paper. The ontology in [Despres and Szulman 2004], in fact, ensures a close relation with the legal text, even though it does not includes axioms which allow shallow reasoning on specific legal phenomena.

The ontology in [Shaheed et al. 2005] is very interesting for the orientation towards NLP, the solid basis on metaphysics, and in that it allows shallow reasoning on a set of simple legal sentences. It is built around the NM ontology ([Shaheed et al. 2005] contains a comparison to LRI-Core), and relies on agents to bridge the legal text with the syntax. The approach is very interesting, yet the focus on agents somewhat overcomplicates the reasoning on complex legal concepts such as that of judicial interpretation. Detecting advanced concepts in legal documents requires in fact a highly complex semantic structure, which prevents the reasoning on a large scale of document contents (for a general account on how to model complex legal concepts for automatic detection see [Palmirani et al. 2011b]). Moreover, as already noted, modelling the dynamics of legal procedure requires a proper implementation of argumentation theory.

3.4.2. A bridge towards judicial argumentation

The argumentation system described in [Ceci and Gordon 2012] allows combining the features of the DL-based ontology with non-monotonic logics such as Defeasible Logics. In particular, Carneades is based on Walton's theory [Gordon and Walton 2009] and also gives account for most of Prakken's consideration on the subject [Prakken 2008] including argumentation schemes and burden of proof. The Carneades application succeeded in performing the tasks of finding relevant precedents, validating the adjudications and suggesting legal rules, precedents, circumstances that could bring to a different adjudication of the claim.

Many projects tried to represent case-law during the nineties, most of which are related to the work of Prof. Kevin Ashley such as [Ashley 2009]. Their main focus is similar to the one of the present research: capturing the elements that contribute to the decision of the judge. The approach was, however, based on concepts rather than on the legal documents themselves. They were meant to teach legal argumentation in law classes. No account for the metadata of the original text was given, and there was no ontology underlying the argumentation trees that reconstruct the judge's reasoning. Rather than representing a single judicial decision, the approach presented in this paper allows instead to connect knowledge coming from different decisions and to highlight similarities and differences between them, not only on the basis of factors, dimensions or values, but also on the basis of the efficacy of the legal documents involved (under criteria of time, hierarchy, and others). Of course, *templatizing* legal documents is a very complex task (see next section, 3.5.1.): the intention, in any case, is not to provide a complete NLP tool but to create an interface through which a legal expert can easily identify the legal concepts evoked by single words, and combinations of them, in legal documents.

Deontic defeasible logic systems, such as those presented in [Nute 1998, Kontopoulos et al. 2011, Governatori and Rotolo 2004] constitute indeed a powerful tool for reasoning on legal concepts. Most of them are explicitly built to import RDF triples, which means that they can perform reasoning on knowledge bases contained in ontologies such as the one presented in this paper. These projects are therefore placed at an upper layer than the one discussed here: the ontology, in the perspective of the present research, should refrain from highly structuring the text syntax and focus only on the document semantics and basic relations, in order to perform shallow reasoning oriented mostly to the data completion, enhanced by the open world assumption. Over a such-built knowledge base, rule systems based on advanced logic dialects (such as those presented in the cited works) could perform highly complex reasoning with tools such as SPINdle (see 6.2.) by importing only the set of triples that best suits their syntactic needs. This should be preferred to approaches that try to extend DL to perform defeasible reasoning such as [Antoniou et al. 2009]:the judicial ontology set showed that it is possible to perform shallow reasoning while staying within OWL2, and in order to perform an efficient reasoning on legal concepts it is not sufficient to implement defeasible reasoning, being also necessary to rely on argumentation schemes [Walton et al. 2008]

The same considerations apply to the approach in [Minh et al. 2009], very interesting in that it provides a simple and intuitive way to encode default knowledge on top of terminological KBs: such a reasoning system does not reach the complexity needed to manage legal concepts (for which deontic defeasible logics are required, with an account for argumentation schemes). This means that a distinct layer is needed in order to perform deep reasoning on the KB: being this the situation, it is better to stay within the achieved standard of OWL2 when performing basic reasoning on KB.

In this perspective, the idea of deriving a closed-world subset of an OWL2 KB as presented in [Ren et al. 2010] seems an optimal enhancement of the present ontology, and will in fact be explored, always keeping in mind, though, that introducing negation-as-failure in OWL2 is not sufficient to grant the ontology layer the expressivity required for performing argumentation tasks.

3.5. Issues

3.5.1. The knowledge acquisition bottleneck

The modelling of the sample ontology library and the extraction of knowledge from the case law sample was carried out manually by a graduated jurist. Also the qualified fragment of text under the Akoma Ntoso standard should be annotated by legal experts: at the present time, manual data insertion seems the only viable choice in the legal domain, as automatic information retrieval and machine learning techniques, do not yet ensure a sufficient level of accuracy (even if some progress in the field has been made, for example in applying NLP techniques to recognize law modifications [Palmirani and Brighi 2010]).

The manual markup of judicial decisions, however, doesn't seem to be sustainable in the long time. For an efficient management of the knowledge acquisition phase, a combination of tools supporting an authored translation of text into semantics should limit the effects of this (still) unavoidable bottleneck: special editor tools (e.g. Norma-Editor) can allow an easy linking between text, metadata and ontology classes, while the more complex ontology constructs (i.e.

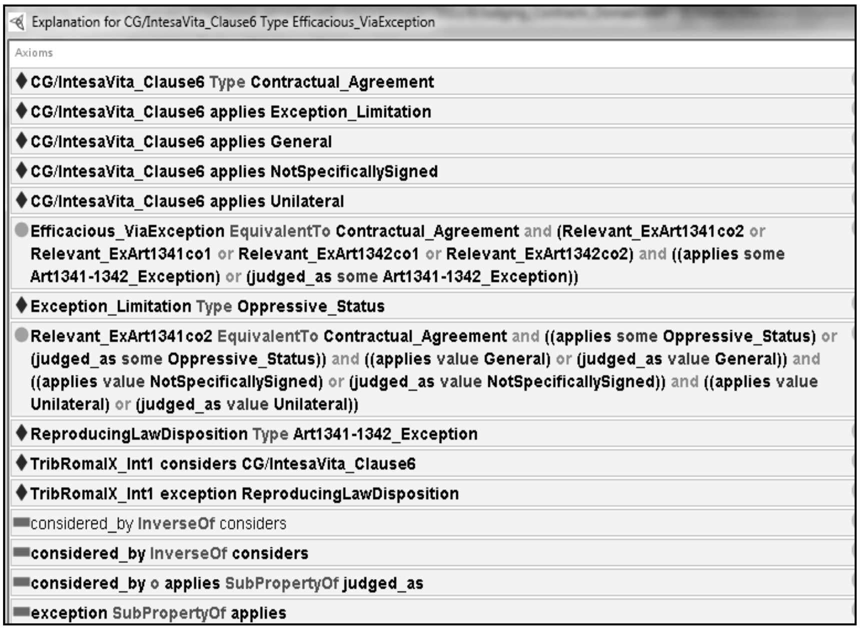


Fig. 3.25 - Explanation of a sample contract clause being not inefficacious because of an exception.

the "considers/applies" constructs) could be managed by an editor plug-in. In this perspective, stronger constraints could be added to the legal core ontology in order to allow these plugin to automatically complete a part of the classification work, leaving to the user the duties of checking and completing the model drafted by the machine.

3.5.2. Representing exceptions

A critical issue in representing the decision's content is represented by exceptions to legal rules. How to model a situation when a material circumstance applies all the legal statuses required by the legal rule, but nevertheless does not fall under that legal rule's legal consequence because it follows some additional rule which defeats the first one? As it should be clear, that issue has no straight solution inside DL, such as OWL-DL logics: introducing some negative condition for the rule to apply (in the form `if (not (exception))`), the open-world assumption OWL relies on would requires to explicitly state for each case that no exception applies. This would hinder the reasoning capabilities of the ontology library explained so far. A solution to this problem could rely on the modelling of the exceptional case as a subclass of the normal case, (see Figure 3.25). In this way, only the instances which are relevant under the law are eligible to be an exception to the application of that law.

This solution has the advantage of allowing reasoning on exceptions without the need to rely on rules. The backside is that the classification of the circumstance as "exceptional" is added to the classification of inefficacy, not substituted to it (Figures 3.26 and 3.27). Again, this issue takes origin from the open world assumption, and cannot be easily avoided while remaining inside OWL-DL: whenever the reasoner is prevented to link a circumstance to a legal consequence, asking him to check that no exception exists, the reasoner will be incapable of inferring anything unless all information concerning the exceptions is explicitly stated in the ontology.

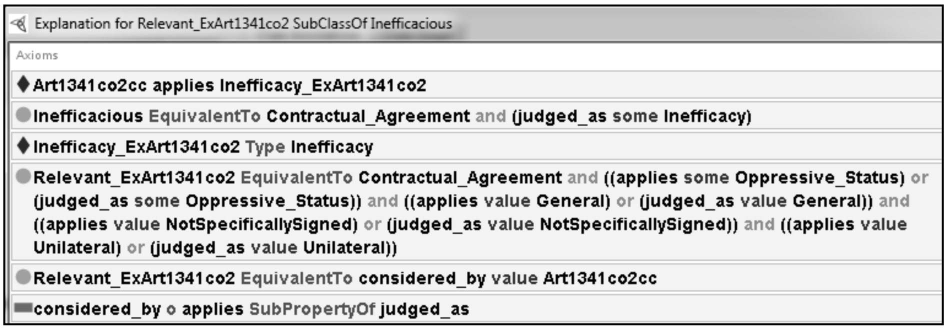


Fig. 3.26 - Explanation for Relevancy being inferred as a subclass of Inefficacious.

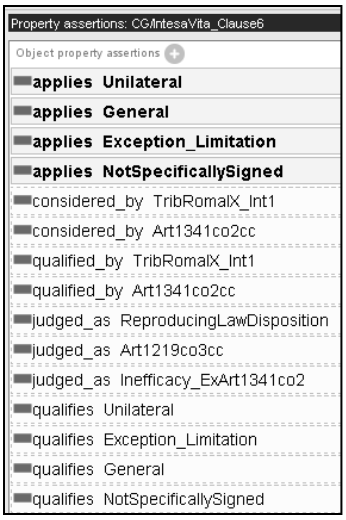


Fig. 3.27 - Stated and inferred property assertions on the "exceptional" contractual agreement.

This issue represents the main reason why a complete syntactic modelling of legal rules is not feasible inside the ontology library, requiring instead a rule system (such as LKIF-Rules [Gordon 2008], Clojure, or LegalRuleML [Palmirani et al. 2011c]) to be fully implemented. Nevertheless, the so-built ontology library represents the ideal background for such a rule system.

3.6. Conclusions

The ontology library presented in this chapter is the pivot of an innovative approach to case-law management, filling the gap between text, metadata, ontology representation and rules modelling, with the goal of detecting all the information available in the text to be enhanced in the legal reasoning through an argumentation theory. This approach allows to directly annotate the text with peculiar metadata representing the hook for the core, domain and argument ontologies. OWL2 is used to get as close as possible to the rules, in order to exploit the computational characteristic of description logics. On the other hand, the ontology framework has a strong weak point in the management of exceptions. It is thus necessary to devolve the deeper legal reasoning features to an upper layer in based on a different logics, namely defeasible logics such as that presented in [Governatori and Rotolo 2004], with added support for argumentation schemes. In the next chapter, an system for argumentation based on the ontology set just presented, and on the Carneades Argumentation System, is described.

Chapter 4

Modelling Judicial Arguments

> "Legal reasoning is not primarily deductive, but rather a modelling process of shaping an understanding of the facts, based on evidence, and an interpretation of the legal sources, to construct a theory for some legal conclusion."
>
> – Jon Bing, *Uncertainty, decisions and information systems*.

> "If you don't like what someone has to say, argue with them."
>
> – Noam Chomsky.

4.1. Introduction

Part of the present research was conducted at the Fraunhofer-FOKUS Institute of Berlin[17]. The aim was the extension of the judicial framework in the rule layer, enhancing the knowledge contained in the core and domain ontologies with a set of rules representing the argumentative processes followed by the judge when interpreting a material circumstance to apply a legal rule to it. To achieve this it is not sufficient to rely on OWL DL reasoning capabilities, but defeasible logics and argumentation patterns are needed.

4.2. Carneades

Carneades[18] is a set of open source software tools for mapping and evaluating arguments, under development by Thomas F. Gordon since 2006. Carneades contains a logical model of argumentation based on Doug Walton's theory of argumentation, and developed in

---

[17] The FOKUS (Fraunhofer Institute for Open Communication System) institute develops solutions for communications in the future. Prof. Thomas F. Gordon (co-tutor of thid Ph.D Thesis) heads a research group on argumentation technology, and is the creator of the Carneades Argumentation System.

[18] http://carneades.berlios.de

collaboration with him. In particular, it implements Walton's argumentation schemes [Walton et al. 2008] not only to reconstruct and evaluate past arguments in natural language texts, but also as templates guiding the user as he/she generates his/her own argument graphs to represent ongoing dialogues. It can therefore be used for studying argumentation from a computational perspective, but also to develp tools supporting practical argumentation processes. The main application scenario of Carneades is that of dialogues where claims are made and competing arguments are put forward to support or attack these claims [Walton 1998], but it also takes into account the relational conception of argument[19] [Dung 1995].

4.2.1. Theoretical background

The focus of the Carneades framework is the modelling of the different techniques used when dealing with argumentation, with a particular attention on dialogical processes: the work relies on Wigmore's charting method (see 1.3.3.2.), Pollock's Framework (see 1.3.4.2.), and Walton's argumentation schemes ([Walton 1996], see following sections). Reflecting these efforts, Carneades proved to be a valid instrument when dealing with fine-tuned schemes about proof standards, burden of proof and when the need arises to distinguish between the logical, dialectical and rhetorical layers [Gordon 2010]. These features, however, are not central for the purposes of the present research: more attention will be focused on the features of evaluating arguments and deriving arguments from ontologies and rules.

4.2.1.1. Argumentation schemes

An argumentation scheme is a pattern of reasoning used in everyday conversation and other contexts, such as legal and scientific argumentation. Argumentation schemes serve the same purpose as their ancestors, the τόποι of Aristotle: they are useful to create, evaluate and classify arguments. Interestingly, the other models that claim to fulfill this ancestral role are S. Toulmin (see 1.3.3.4.) and D. Schum (see 1.3.3.6.). In recent times, however, the Artificial Intelligence field has become increasingly interested in argumentation schemes, due to their potential for making improvements in the reasoning capabilites of agents [Garssen 2001; Bex et. al. 2003; Verheij 2003].

In argumentation theory, argumentation schemes are evaluated through a set of critical questions (CQs), specific for each scheme. Critical questions were first introduced by Arthur Hastings [Hastings 1963] as part of his analysis on presumptive argumentation schemes. Each question reveals possible weak points in the argumentation, and if not answered adequately may render that specific argument useless in supporting the speaker's position in the dialogue. Evidently, critical points in arguments should be formalized in a dialogical structure, in order to maintain the notion of defeasibility of every argument in the scheme, including those introduced to answer one or more critical questions. The following example will better explain this dialogical structure.

[19] The main difference between the two conceptions of argument is that a proposition which has not been attacked is acceptable in the relational model of argument, while in most dialogues it would be not acceptable, since in most schemes making a claim involves having the burden of proof on it.

Argument from expert opinion is a classic example of how argumentation schemes and critical questions can help evaluating the validity of a specific argument: it is a kind of argument which is generally considered as reasonable, but also brought to major fallacies in past cases. In Walton's analysis [Walton 1996], expert opinion is represented by the following scheme:

Source E is an expert in domain D

E asserts that proposition A is known to be true (or false)

A is within D

Therefore, A may plausibly be taken to be true (or false).

As shown by experiments in social psychology, there is a tendency to defer to experts, sometimes without questioning, resulting in fallacious appeals to authority. Many circumstances could prevent the apparently deductive conclusion that "if E says A, then A is true": in particular, epistemic closure in an expert field is far from truth, and therefore an expert can never be considered as knowing everything in a domain, and neither can its opinion be deductively true beyond challenge. Thus for many (if not all) appeals to the expert opinion, the deductivist approach does not work. As [Reed et. al 2004] puts it, critical questions are used to ease tensions between forms of argument that are clearly reasonable in some instances, but that cannot be analysed as deductively valid.

[Walton 1997] identifies six basic critical questions matching the appeal to expert opinion:

- How credible is E as an expert source?
- Is E an expert in D?
- Does E' testimony imply A?
- Is E reliable?
- Is A consistent with the testimony of other experts?
- Is A supported by evidence?

Please notice that, in many cases, asking one of the basic critical questions above will lead to critical subquestion at a deeper level of examination. This is one way to create argumentation graphs (see 4.2.2.1.).

4.2.1.2. Premise evaluation: burden of proof

According to [Walton 2005], the method for evaluating an argument through critical questions is by a shifting of burden of proof in a dialogue: when the proponent arguments his position with an expert opinion and the respondent poses one of the six critical questions, a burden of proof shifts back to the proponent's side, defeating or undercutting the agument temporarily until the critical question has been answered successfully. Please notice that the notions of *burden of proof* and *proof standards* (explained later) are relevant only when argumentation is viewed as a dialogical process for making justified decisions, and therefore they will not be thoroughly exploited in the present research. Nevertheless, also these features of Carneades will be fully explained, for two reasons: first, presenting the capabilities and the theories beyond Carneades is impossible without referring to these notions; second, these notions play a role – even if a limited one – in the Carneades extension of the Judicial Ontology Set.

There are differences between the critical questions on how strongly they produce that shift. Studies in the field highlighted two main theories about the allocation of the burden of proof when critical questions are asked [Walton 2005]: according to one theory, the shift is automatic and the argument is defeated unless the proponent provides an appropriate answer, while according to the other theory the shift occurs only if the critical question is backed up with some evidence. Verhej [Verheij 2003] distinguishes four types of critical questions, applying them to his Deflog logic (see 1.2.7.1.).

a. CQs used to question whether a premise of a scheme holds (redundant);

b. CQs pointed to exceptional situations in which a scheme should not be used (undercutters);

c. CQs setting conditions for the proper use of a scheme (denying assumptions);

d. CQs pointing to other arguments that might be used to attack the scheme (rebuttals).

This classification explains why some critical questions shift the burden of proof back to the proponent while others must be backed up with evidence. In particular, the difference between these types of CQs comes into play when dealing with a lack of response from the other party. Under this point of view, not all argumentation dialogues are the same: in some cases, silence implies consent to the major and minor premise, and in this perspective the CQ of type *a* is not redundant, but rather the dialogical instrument used to ask for the proof of a premise. Delving deeper into the analysis of the premises being attacked by the CQ, it turns out that the type of premise directly determines the type of CQ being posed: When pointing to exceptional situations as in *b*, the CQ is targeting an exception and trying to prove its existence in the present case: in this case, the burden of proof does not shift back to the proponent. When setting conditions for the proper use of the scheme as in *c*, the respondent is trying to falsify one of the assumptions in the proposer's argument[20]. Finally, when the respondent evokes other, incompatible arguments to attack the scheme, he has the burden of proof of proving the acceptability of the argument, which will then count as a premise towards the rebuttal of the proponent's argument. While 'static', relational AI models of argumentation cannot handle the distinctions between normal premises, exceptions and assumptions (since these distinctions only make sense in a procedural context), Carneades is specifically designed to successfully handle the allocation of the burden of proof, especially in legal domain.

The distinction presented above leads the main intuitions about the allocation of the burden of proof: when a party provides an argument instantiating some applicable scheme, the burden is

[20] Please note that assumptions are not the same as negation of exceptions, thus the use of the term *assumption* in Carneades' background theory (see 4.2.2.2.) deviates, e.g., from [Bondarenko, et al. 1997] (see 1.3.1.2.), where it is used for negations of what we call exceptions.

on that party to prove its ordinary premises and (once challenged) its assumptions, after which the burden shifts to the other party to defeat the argument by rebutting it or pointing out exceptions.

4.2.1.3. Burden of production and burden of persuasion

In civil law suits of common law, it is normally the plaintiff who has the burden of proof for the main claim. However, Wigmore [Wigmore 1940] distinguished two types of burden of proof: the burden of persuasion, and the burden of production: "The risk of non-persuasion operates when the case has come into the hands of the jury, while the duty of producing evidence implies a liability to a ruling by the judge disposing of the issue without leaving the question open to the jury's deliberations" ([Wigmore 1940], pp. 285-286). Wigmore says that the burden of persuasion never shifts, while the duty of producing evidence to satisfy the judge does have this shifting characteristic. Also McCormick [Strong 1992] and Park, Leonard and Goldberg [Park et al. 1998] provide a similar distinction between burden of persuasion and burden of production. The two burdens can be distributed over the two parties: for example, the plaintiff usually has the burden of production for the premises of the main claim, while the defendant has the burden of production for exceptions.

Efforts towards a modelling of the burden of persuasion are contained in [Gordon and Walton 2009] and in Prakken and Sartor's non-monotonic system [Prakken and Sartor 1997] based on Dung's grounded semantics (see 1.3.2.1.), as adjusted in [Prakken 2001] to distinguish between cases where the exception merely has to be made plausible and cases where it has to be proven. The modified system assumes as input not just a set of rules but also an allocation of proof burdens to plaintiff and defendant, which assume different dialectical roles (proponent or opponent) for different propositions. Although Prakken does not explicitly distinguish the burdens of production and persuasion, Prakken and Sartor [Prakken and Sartor 2006] argue that this work in fact models distribution of the burden of persuasion.

Carneades allows allocation of the burden of production and the burden of persuasion separately, during the course of the dialogue. The initial allocation of the burden of production is regulated by the premise types: evidence for ordinary premises and assumptions (once challenged) must be produced by the proponent of the argument, while evidence for exceptions must be produced by the respondent. After the burden of production has been met, the burden of persuasion can be distributed by adjusting the proof standard required by the premise. The distinction of statements between exceptions, assumptions and ordinary premises is done by matching the argument instance with an argumentation scheme. As the dialogue progresses, the burdens may be reallocated either by matching some burden of proof or by changing the assignment of premise types and proof standards via additional speech acts, each of which must be matched to the the appropriate argumentation scheme.

4.2.1.4. Proof standards

According to the definition of [Gordon et al. 2009], the *proof standard* - a central concept in the dialogical approach to argumentation- is a function mapping tuples of the form <issue, stage, audience> to the Boolean values *true* and *false,* where:

– **Issue** is a proposition in a propositional language;

– **Stage** is a tuple <arguments, status> where *arguments* is a set of arguments (see next section) and *status* is a function mapping the conclusions of the arguments to their dialectical status in the stage (possible statuses are claimed, questioned, accepted, rejected);

– **Audience** is a structure <assumptions, weight> where *assumption* is a consistent set of literals in a propositional language, assumed to be acceptable by the audience, and *weight* is a partial function mapping arguments to real numbers, representing the relative weights assigned by the audience to the arguments.

Thus, a literal *p* is acceptable in an argument evaluation structure <*stage*, *audience*, *standard*>, if and only if *standard*(*p*, *stage*, *audience*) is *true*.

These are the proof standards currently defined in Carneades, from [Gordon et. al. 2009, Brewka et. al. 2010]:

– **Scintilla of Evidence** *(se)*: it is satisfied iff there is at least one applicable argument *pro p*;

– **Preponderance of Evidence** *(pe)*, also called *best argument* in [Gordon and Walton 2006, Gordon et. al. 2007]: it is satisfied iff *p* satisfies *se* and the maximum weight assigned to an applicable argument *pro p* is greater than the maximum weight assigned to an applicable argument *con p*;

– **Clear and Convincing Evidence** *(ce)*: it is satisfied iff *p* satisfies *pe* and the maximum weight of applicable pro arguments exceeds some threshold α, and the difference between the maximum weight of the applicable *pro* arguments and the maximum weight of the applicable *con* arguments exceeds some threshold β;

– **Beyond Reasonable Doubt** *(bd)*: it is satisfied iff *p* satisfies *ce* and the maximum weight of the applicable *con* arguments is less than some threshold γ;

– **Dialectical Validity** *(dv)*: it is satisfied iff there is an applicable argument *pro p* (*p* satisfies *se*) and there is no argument *con p*.

Please notice that the first four proof standards are ordered by their relative strength, from the weakest to the strongest; Dialectical Validity, however, cannot be easily placed into that list, since it does not give importance to the weight of the *pro* argument(s).

This list doesn't claim to be exhaustive, nor to adequately model legal proof standards. In addition, a proof standard can also be derived from another by switching the roles of *pro* and *con* arguments of an existing standard: it is called the *complement* of a proof standard [Gordon et. al. 2006; Gordon et. al. 2007]. For example, the complement of the *pe* standard is satisfied iff the statement is supported by some applicable *con* argument with priority over its strongest applicable *pro* argument. In principle a statement can satisfy both a proof standard and its complement, highlighting the difference between acceptability and truth (see 4.2.2.2.): the arguments can be sufficient to justify a decision either way without being inconsistent. In practice, however, this will be the case only with very weak proof standards, such as scintilla of evidence: if there exists an applicable *pro* argument and an applicable *con* argument, then both *se* and its complement are satisfied.

4.2.2. Argument structure

On the basis of the theories explained above, argument structure in Carneades can be explained through a set of definitions contained in [Gordon 2006]. The atomic element of arguments is the stamement:

> **Definition of Statement –** Let <statement = complement> be a structure, where *statement* denotes the set of declarative sentences in some language, = is an equality relation, modeled as a function of type "statement x statement → Boolean", and *complement* is a function of type "statement→ statement" mapping a statement to its logical complement. If *s* is a statement, the complement of s is denoted ¬*s*.

Next, three types of premises are distinguished to allow defeasible argumentation and distribution of the burden of proof (as seen above):

> **Definition of Premise –** Let *premise* denote the set of premises. There are the following types of premises:

- If *s* is a statement, then ◊*s*, called *ordinary premise*, is a premise.
- If *s* is a statement, then ○*s*, called *assumption*, is a premise.
- If *s* is a statement, then ●*s*, called *exception*, is a premise.
- Nothing else is a premise.

These elements form the structure of arguments:

> **Definition of Argument –** An *argument* is a tuple <c, d, p>, where *c* is a statement, $d \in$ {pro, con} and $p \in 2^{\text{premise}}$. If *a* is an argument <c, d, p>, then conclusion(*a*) = *c*, direction(*a*) = *d* and premises(*a*) = *p*.

The main distinction between arguments and inference rules is the definition of *con* arguments: semantically, these are instances of inference rules for the negation of the conclusion. This is an approach which abstracts from the syntax of the language for statements, in order to be able to aggregate arguments *pro* and *con* some statement and to resolve the conflicts.

4.2.2.1. Argument graphs

Carneades represents argumentation dialogues in tree-like argument graphs, whose conception is similar to Pollock's concept of an *inference graph* (see 1.3.4.2.), where there are nodes representing statements (propositions) and links indicating inference relations between statements. Unlike Dung's model (see 1.3.2.1.), in which the internal structure of single arguments is irrelevant, Carneades' model of the acceptability of statements is based on the concept of argumentation schemes, where each conclusion is linked to a set of premises which, in turn, may be accepted or rejected. On the other hand, the syntax of single statements is not relevant, except for the identification of the logical complement of a statement: a statement and its complement are in fact, unlike all other statements, related. A dispute about *s* is also a dispute about ¬*s*: an argument pro one is an argument con the other, and if one of these statements is accepted the other is automatically rejected. For these reasons, both statements are represented with only one statement node in the argument graph.

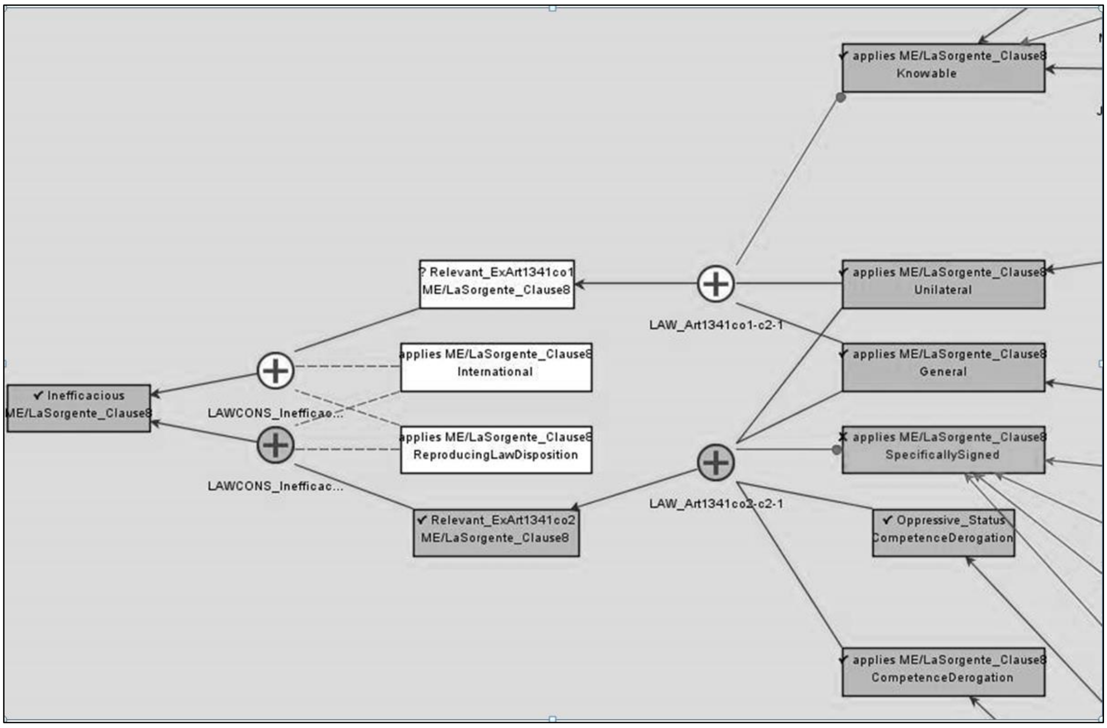


Fig. 4.1 – Example of argument graph (visualized with Carneades 1.0.2). Green boxes represent accepted statements (or acceptable and complement non acceptable), red boxes represent rejected statements (or non-acceptable and complement acceptable), white boxes represent undecided statements.

[Walton 1996] identifies two kinds of arguments: *convergent arguments*, where each reason provided (premise) is sufficient to accept the conclusion, and *linked arguments*, where all premises must hold for the conclusion to be accepted. Convergent arguments are represented in the graph by building multiple arguments for the same conclusion, while linked arguments are represented by a single argument defended by a set of premises instead of a single one.

Assumptions and exceptions can be used to model Walton's concept of critical questions [Walton 1996b], as already explained.

Please note that argument graphs are not restricted to trees, as the same statement can be used in different types of premises in several arguments. However, graphs are not completely general: in order to enhance the decidability of the acceptability property of statements, in Carneades

1.0.2 cycles are not allowed[21]. Even if limitations can arise from this restriction, some of them can be successfully tackled: for example, in systems using Dung's approach most cycles are constituted by clashes between arguments with incompatible conclusion, which can be represented in Carneades as a pair of arguments pro and con the same statement, without introducing cycles.

4.2.2.2. Argument evaluation

In Carneades, argument evaluation consists in determining whether a statement is *acceptable* in an argument graph. The *acceptability* relation between argument graphs and statements measures the sufficiency of the proof: it is accomplished if and only if the argument graph is a defeasible proof of the statement. This distinguishes the acceptability relation from the defeasible consequence relation of non-monotonic logics: assuming a correct and complete calculus for such logics, a statement is a defeasible consequence of a set of statement if and only if the statement is *derivable* from the calculus, whether or not the proof has actually been *derived* in the specific case or not.

Intuitively, a statement is acceptable if it is true under the arguments which have been put forward in the dialogue. Acceptability of a statement, as already explained, depends on its proof standard. Satisfiability of the proof standard depends on the defensibility of the arguments *pro* and *con* the statement, which in turn can be derived from the acceptability of the premises of the argument. This behaviour is called *recursiveness*.

To evaluate a set of arguments in an argument graph, three factors have to be known:

- The current dialectical status of each statement in the dialogue (stated/questioned/accepted/rejected). This status can be derived pragmatically, from the speech acts which constitute the dialogue;
- The proof standard assigned to each statement;
- A strict partial ordering on arguments (*a1>a2* means that *a1 has priority over/is stronger than a2*).

These requrements are formalized in the following definition of argument context, contained in [Gordon 2006]:

> **Definition of Argument context –** Let *C*, the argument context, be a tuple (status, ps, >), where *status* is a function of type "statement→{stated, questioned, accepted, rejected}", *ps* is a function of type "statement→proof-standard" and > is a strict partial ordering on arguments. For every statement *s* and its complement ¬*s*, the proof standard assigned to ¬*s* is the complement of the proof standard assigned to s and:

[21] This issue has been tackled in the new Carneades Policy Modelling Tool, following [Brewka and Gordon 2010] and [Van Gijzel and Prakken 2011].

– if status(s) = *stated* then status(¬s) = *stated,*

– if status(s) = *questioned* then status(¬s) = *questioned,*

– if status(s) = *accepted* then status(¬s) = *rejected,*

– if status(s) = *rejected* then status(¬s) = *accepted*.

Please note that stating or questioning a statement does not imply the assertion of some position or viewpoint *pro* or *con* the statement: stating a statement merely introduces it into the dialogue, while questioning a statement merely makes an issue out of it.

Definition of Acceptability of statements – Let *acceptable* be a function of type "statement x argument-graph→boolean". A statement *s* is *acceptable* in an argument graph *G* if and only if it satistfies its proof standard: "acceptable (s, G) = satisfies (s, ps(s), G)"

Definition of Satisfaction of proof standards – A proof standard is a function of type "statement x argument-graph→boolean". A statement *s* is *satisfied* by a proof standard *f* in an argument graph *G* if and only if *f* (*s*, *G*) is true.

Definition of Defensibility of arguments – Let *defensible* be a function of type argument x argument-graph→boolean. An argument *α* is defensible in an argument graph *G* if and only if all of its premises *hold* in the argument graph: "defensible (α, G)=all(λp.holds(p,G))(premises α)".

Definition of Holding of premises – Let *holds* be a function of type *premise* x *argument-graph→boolean*. Whether or not a premise holds depends on its type. Thus, there are the following cases:

– If *p* is an ordinary premise, ◊*s*, then

$$\text{Holds (p, G)} = \begin{cases} \textit{acceptable}\ (s, G) & \text{if status(s) = } \textit{stated} \\ \textit{acceptable}\ (s, G) & \text{if status(s) = } \textit{questioned} \\ \textit{true} & \text{if status(s) =} \textit{accepted} \\ \textit{false} & \text{if status(s) =} \textit{rejected} \end{cases}$$

– If *p* is an assumption, ●*s*, then

$$\text{Holds (p, G)} = \begin{cases} \textit{true} & \text{if status(s) = } \textit{stated} \\ \textit{acceptable}\ (s, G) & \text{if status(s) = } \textit{questioned} \\ \textit{true} & \text{if status(s) =} \textit{accepted} \\ \textit{false} & \text{if status(s) =} \textit{rejected} \end{cases}$$

– If *p* is an exception, ○*s*, then

$$\text{Holds }(p, G) = \begin{cases} \neg \textit{acceptable}\ (s, G) & \text{if status(s) = } \textit{stated} \\ \neg \textit{acceptable}\ (s, G) & \text{if status(s) = } \textit{questioned} \\ \textit{false} & \text{if status(s) =} \textit{accepted} \\ \textit{true} & \text{if status(s) =} \textit{rejected} \end{cases}$$

Please notice that it can be proven that there is always a unique and complete assignment of ‘acceptable’ or ‘not acceptable’ to statements, and of ‘holds’ or ‘not holds’ to premises [Gordon and Walton 2006, Gordon et al. 2007]. It is also important to notice that whether or not a premise holds depends not only on the arguments which have been put forward, but also on the type of premise and the status of the premise’s statement in the context, which progresses in the course of the dialogue: first they are introduced (*stated*), and whether or not a premise which uses this statement holds depends on the kind of premises (ordinary premises hold only if the statement is acceptable given whatever arguments have been put forward, assumptions hold unconditionally, and exceptions hold only if the statement is not acceptable given the arguments put forward); successively a statement can be questioned, becoming an issue: now both ordinary premises and assumptions which use this statement hold only if they are acceptable; finally, a decision (whose justification – given the arguments made and the applicable proof standards – can be checked by anyone interested) accepts or rejects some statement, and the model respects that decision: for example, ordinary premises with accepted statements hold and ordinary premises with rejected statements do not hold.

4.2.3. Rules

The kind of rule that Carneades and the AI&Law community is interested in is that defined as “One of a set of explicit or understood regulations or principles governing conduct within a particular sphere of activity”[Jewell and Abate, 2001]. The argumentation scheme for *argument from legal rules* which constitutes the background theory for the Carneades rule system, contained in [Gordon 2010], is based on the one developed by Gordon in The Pleadings Game [Gordon 1995] and can be defined formally as follows:

Argument from legal rules

Premises:

r is a legal rule with ordinary conditions *a(1),...,a(n)* and conclusion *c*;

Each *a(i)* in *a(1)...a(n)* is presumably true.

Conclusion: *c* is presumably true.

Critical Questions:

Does some exception of *r* apply?

Is some assumption of *r* not met?

Is *r* a valid legal rule?

Does some rule excluding *r* apply in this case?

Here is a simplified example of a Carneades-style rule, built to represent a part of the knowledge already present in the ontology set:

```
(rule Art1341co2
      (if   (and (oppressive ?c)
                 (general ?c)
                 (unilateral ?c)
                 (unless (specificallysigned ?c)))
            (inefficacious ?c)))
```

Following is the list of characteristics that can be ascribed to that kind of rule, according to the AI&Law community [Gordon 2005, Prakken et. al. 1996, Hage 1997, Verheij 1996]:

– Rules have properties – such as their date of enactment, jurisdiction or authority;

– When the antecedent of a rule is satisfied by the facts of a case, the conclusion of the rule is only presumably true, not necessarily true;

– Rules are subject to exceptions;

– Rules can conflict;

– Some rule conflicts can be resolved using rules about rule priorities;

– Exclusionary rules provide one way to undercut other rules;

– Rules can be invalid or become invalid. Deleting invalid rules is not an option when it is necessary to reason retroactively with rules which were valid at various times over a course of events;

– Rules do not counterpose. If some conclusion of a rule is not true, the rule does not sanction any inferences about the truth of its premises.

Because of the first characteristic, rules cannot be modeled adequately as material implications in predicate logic: they need to be reified as terms, not formulas, so as to allow their properties to be expressed and reasoned about for determining their validity and priority.

Like all monotonic logics, DL – although very powerful and useful – is not sufficient for modelling legal rules, such as the rules of contract law, in way which is maintainable, verifiable and isomorphic with the structure of legal texts (see [Gordon 1986, 1987, 1988]). This is mainly because legislation is tipically constituted by general rules subject to exceptions. Therefore, arguments made by applying legal rules are defeasible: their conclusions can be defeated with better counterarguments. Moreover, legal rules may conflict with each other, and these conflicts are resolved using higher-level rules, defining the priority relationships between rules such as the rules of *lex superior*, *lex posterior*, *lex specialis*. Here is a first example of a rule, representing the legal principle that “later rules have priority of earlier rules”:

```
(rule lex-posterior
      (if (and (enacted ?r1 ?d1)
            (enacted ?r2 ?d2)
            (later ?d2 ?d1))
         (prior ?r2 ?r1)))
```

Rules can be defeated in two other ways: by challenging their validity or by showing that some exclusionary condition applies. These are modeled with rules about validity and exclusion, using two further built-in predicates: `(valid <rule>)` and `(excluded <rule> <literal>)`, where `<rule>` is a constant naming the rule and `<literal>` is a compound term representing a literal, which thus can also be reified in this system [Gordon 2008]. Also the *valid* and *excluded* relations, like the *prior* relation, are to be defined in models of legal domains. Here are two examples, one for each of these predicates:

```
(rule unconstitutional
      (if  (unconstitutional ?r1)
            (not (valid ?r1))))
(rule Art.1341co2-i
      (if  (specifically_signed ?c)
            (excluded Art.1341 (inefficacious ?c)))
```

In the Carneades software, expertise from logic programming has been adapted to model legal rules and build an inference engine which can construct arguments from rules (see below 4.2.10.1.). Rules in logic programming are Horn clauses (see 1.2.2.3.). Since it is not possible to represent negative facts using Horn clauses, rules do not counterpoise in logic programming. In Carneades, both the head and the body of the rules are more general than they are in Horn clauses: the head consists of a set of (both positive and negative) literals, while the body consists of an arbitrary first-order logic formula, except that quantifiers and bi-conditionals are not supported.

For these reasons, legal rules are modeled in Carneades using a defeasible rule language specifically designed for this purpose as part of LKIF (see 2.4.5), and use OWL (see 2.2.4.1.) for the only purposes of declaring the language of predicate symbols and to make assertion about these predicates using DL axioms representing knowledge which is universally true and beyond dispute in the domain. The Carneades version described in [Gordon 2011b] is able to import and export both arguments and rules in LKIF format. Carneades version 2.x brings changes in both language and syntax for rules: XML language and LKIF-Rules syntax will in fact be replaced by s-expressivity and Horn Clause (Lisp-like) syntax.

4.2.4. Argument construction

The argument construction module relies on rules, ontologies and manually inserted statements to construct an argumentation tree trying to answer any query concerning a precedent contained in the knowledge base [Gordon and Walton 2009]. The target statement can be of two types: a query-like statement (i.e. *?x Oppressive_Clause*, which means "provide all x, where x is an oppressive clause") or a simple assertion (*ME/LaSorgente_Clause8 Oppressive_Clause*, which means "Clause 8 of the contract between M.E. and La Sorgente is an oppressive clause"): in the first case, the system will return a list of results and arguments, while in the second case the system will construct a single argumentation tree *pro* or *con* the desired goal.

4.2.4.1. Argument from Rules

Carneades compiles rules into clauses in disjunctive normal form (see [Gordon 2007a]). Given an atomic proposition *P*, a rule can be used to construct an argument *pro* or *con P* if *P* or ¬*P*, respectively, are present in the head of the rule: if the head of a rule is an atomic sentence *s*, then the rule is mapped to a scheme for arguments *pro s*. If it is a negated atomic sentence (¬*s*), then the rule is mapped to a scheme for arguments *con s*.

The burden of proof for an atomic proposition in the body can be allocated to the opponent by declaring the proposition to be an exception. Similarly, it is possible to make a proposition assumable, without proof, until questioned.

The next step is to create argumentation schemes from the rules:

> **Definition of scheme for arguments from rules:** Let *r* be a rule, with conditions *a(1)...a(n)* and conclusions *c(1)...c(n)*. Three premises, implicit in each rule, are made explicit here. The first, ○*v*, where *v*=(not(valid *r*)), excepts *r* if it is an invalid rule. The second, ○*e*, where *e*=(excluded *r c(i)*), excepts *r* if it is *excluded* with respect to *c(i)* by some other rule. The third, ○*p*, where *p*=(priority *r(2) r*), excludes *r* if another rule, *r(2)*, exists of higher priority than *r* which is applicable and supports a contradictory conclusion. For each *c(i)* in *c(1)...c(n)* of *r*, *r* denotes an argumentation scheme of the following form, where *d* is 'pro' if *c(i)* is an atomic sentence and 'con' if *c(i)* is a negated atomic sentence:
>
> $$\frac{\mathrm{P(a(1))...p(a(n)),\ \circ v,\ \circ e,\ \circ p}}{\mathrm{d\ c(i)}}$$

In order to construct instantiations of these argumentation schemes, the variables must be systematicallty renamed using a *substitution environment* (a mapping from variables to terms constructed by matching the conclusion of the argumentation scheme with some goal atomic statement). The *valid* and *excluded* relation, as well as the priority rules, must be defined in the models of the legal domains (see above, 4.2.3.).

The *applies* predicate (which has nothing to do with the `applies` OWL property presented in 3.2.3.1.) is a 'built-in', meta-level relation which cannot be defined directly in rules. It is defined in [Gordon 2008] as follows:

> **Definition of the *applies* predicate:** Let σ be a substitution environment and *G* be an argument graph. Let *r* be a rule and *S* be the set of argumentation schemes for *r*, with all of the variables in these schemes systematically renamed. There are two cases, for atomic

literals and negated literals. The rule *r* applies to a literal *P* in the structure < σ, *G*>, if here exists a *pro* argumentation scheme *s* in *S*, if *P* is atomic, or a *con* argumentation scheme, if *P* is negated, such that the conclusion of *s* is unifiable with *P* in σ, and every premise of *s*, with its variables substituted by their values in the σ, *holds* in *G*.

This definition of the *applies* predicate enables some meta-level resoning: it allows to find rules which can be used to generate defensible *pro* and *con* arguments for some goal statement, or to check whether a particual rule can be used to generate them. Please note that the semantics of negation is dialectical, not classical negation or *negation-as-failure*, as the closed-world assumption is not made (see 2.2.5.1.): in Carneades, a negated sentence is acceptable just when the complement of the proof standard assigned to the sentence is satisfied, where the complement of a proof standard is constructed by reversing the roles of *pro* and *con* arguments (see above, 4.2.2.)

The Carneades inference engine uses rules to construct and search a space of argument *states*, where each state consists of [Gordon 2010]:

- **Topic**, the statement which constitutes the main issue of the dialogue;
- **Viewpoint,** either *pro* or *con*. Depending on the viewpoint, the *state* of the argument can satisfy the goal state if the topic satisfies or doesn't satisfy, respectively, the proof standard. Please notice the asymmetry between *pro* and *con*: the *con* viewpoint need not prove the complement of the topic, but only prevent the *pro* viewpoint from achieving its goal of proving the topic;
- **Pro-goals**, a list of disjoint clauses, each of which represents a set of statements which might help the proponent to prove the topic;
- **Con-goals**, a list of disjoint clauses, each of which represents a set of statements which might help the opponent to prevent the proponent from proving the topic;
- **Arguments**, a graph representing all the arguments which have been put forward;
- **Substitution**, a substitution environment mapping schema variables to terms (presented earlier on this section);

– **Candidates**, a list of candidate arguments, which are added to the argument graph only after all of its schema variables are instantiated in the substitution environment, to ensure that all statements in the graph are ground atomic formulas.

Carneades is implemented in a modular way, which allows the space of states to be searched using any heuristic search strategy, many of which have been implemented (such as depth-first search, breadth-first, iterative-deepening). For all the strategies a resource limit, restricting the number of states which may be visited in the search space, may be specified, to assure termination of the search procedure. The system is also extensible.

4.2.4.2. Argument from Ontology

Carneades 1.0.2 allows to import of both the TBox and the ABox of OWL/RDF knowledge bases. The import, however, does not keep the inferential potentiality of the ABox, which would be possible by translating the axioms into LKIF rules, but rather imports a series of single statements about the knowledge base, not distinguishing between asserted and inferred knowledge (which is obtained by using a Pellet reasoner). This is not only a technical choice, but it also has to do with the role played by ontologies. It can be argued that arguments from ontologies should not be defeasible, since ontologies are typically defined using some subset of first-order logic, which is of course monotonic. One might claim that all communication presumes a shared ontology which is not subject to debate. [Gordon et al. 2009] consider arguments from ontology as defeasible, to the same extent as arguments from theory are defeasible. Even if one accepts that a community in principle shares some ontology, this does not imply that a model of this ontology in some representation language, such as OWL, is adequate or beyond dispute, and also when a community or some institutional authority has declared or accepted the ontology as binding, arguments from such agreements and authorities

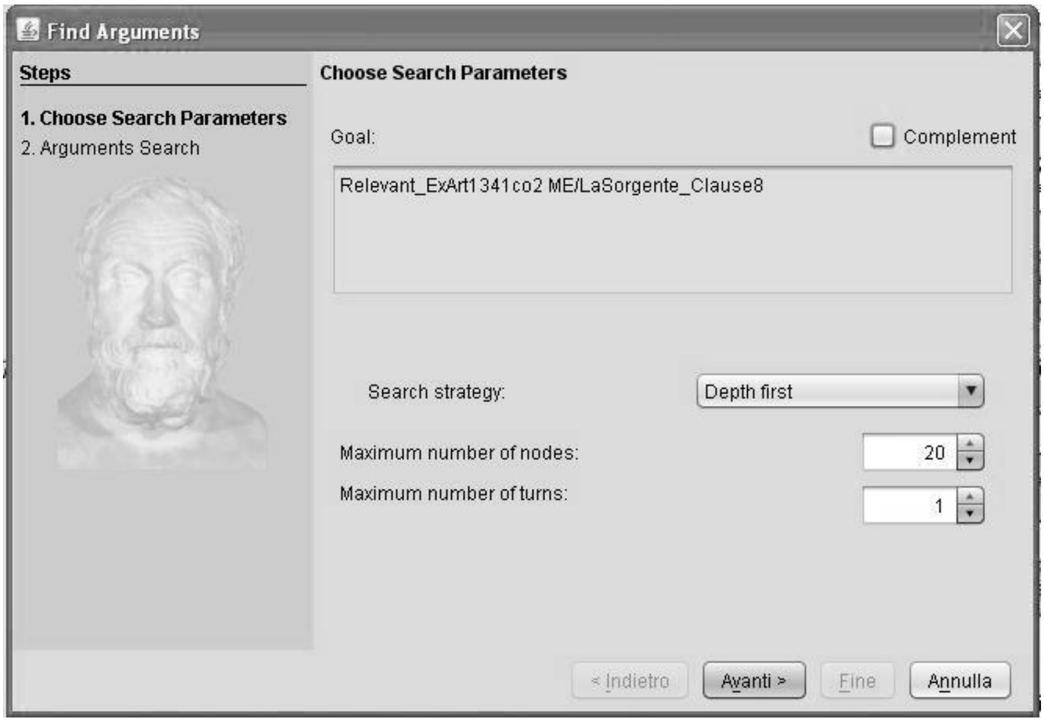


Fig. 4.2 – The argument search dialogue box, with nodes restrictions.

remain subject to critical questions. One way to represent this is to consider argument from ontology as a defeasible argumentation scheme: Walton [Walton 2006] defined a scheme called Argument from verbal classification, which can be considered as a kind of *argument from ontology*.

Argumentation scheme for Argument from Verbal Classification

Individual Premise:

a has property F.

Classification Premise:

for all *x*, if *x* has property F, then *x* can be classified as having property G.

Conclusion:

a has property G.

Critical Questions:

a. Does *a* definitely have property F, or is there room for doubt?

b. Can the verbal classification (in the second premise) be sent to hold strongly, or is it one of those weak classifications that is subject to doubt?

Also the scheme of argument from theory can be used:

Argumentation scheme for Argument from Theory

Derivability premise:

$T \vdash P$

Theory premise:

T is a coherent theory of the intended domain.

Conclusion:

P

Critical Questions:

a. Even though P is necessarily true if T is true, the argument can be challenged by questioning the theory premise. Is the theory T really coherent?

In Carneades 1.0.2, this is translated into a built-in presumption which comes along with every argument coming from ontology: this presumption is the validity of the ontology being used as a source of arguments. It is therefore not possible to examine which axioms of the ontology are involved in the inferencing, and neither to distinguish between asserted and inferred knowledge. This also means (see below) that no back-feeding of new information into the axioms of the Abox is possible.

4.2.4.3. Arguments from cases and testimonial evidence

As already seen in 1.4.2., despite the several research project in the field of Artificial Intelligence and Law on this subject, computational models of case-based reasoning are still at an early developmental stage, used mostly for legal teaching purposes, and the same can be said about a common theory of legal reasoning with cases in legal philosophy.

Carneades implements a reconstruction of Cato (see 1.4.2.2.) by [Wyner and Bench-Capon 2007]. For some legal issue, the precedent cases are analysed to collect the set of *factors* (a

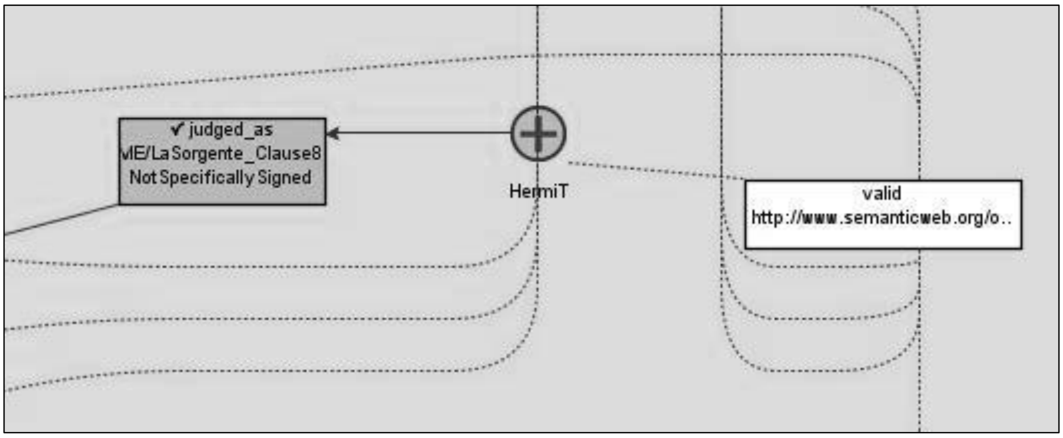


Fig. 4.3 – Presumptions bringing forth the argument from ontology.

proposition which tends to favor one of the parties) which were found relevant for deciding the issue. Each precedent case is modeled as a set of such factors together with the decision of the court regarding the main claim. Starting from the set of factors known (or assumed) to be true in the current case, Carneades constructs (for each precedent) a set of six partitions of the union of the factors of the current case and the precedent case:

- Intersection of pro-*plaintiff* factors in the *precedent* case and in the *current* case;
- Intersection of pro-*defendant* factors in the *precedent* case and in the *current* case;
- Set of pro-*defendant* factors in the *precedent* case which are not in the *current* case;
- Set of pro-*plaintiff* factors in the *precedent* case which are not in the *current* case;
- Set of pro-*defendant* factors in the *current* case which are not in the *precedent* case;

– Set of pro-*plaintiff* factors in the *current* case which are not in the *precedent* case.

Unfortunately, this reconstruction of CATO does not allow sufficient integration of arguments from ontologies or rules, since ontologies and rules are modeled at a finer level of granularity (using predicate logic) than the factors in this model of case-based reasoning, which are at a more abstract, propositional level. This problem can be overcome by using bridging rules (similar to the output/input transformers in [Prakken 2008]), to map predicate logic formulas to factors. However, since different instantiations of schema variables in the predicate logic formulas can get mapped to the same factor, this solution only allows to map at most one predicate logic formula to each factor [Gordon 2010].

4.2.4.4. Argument Visualization

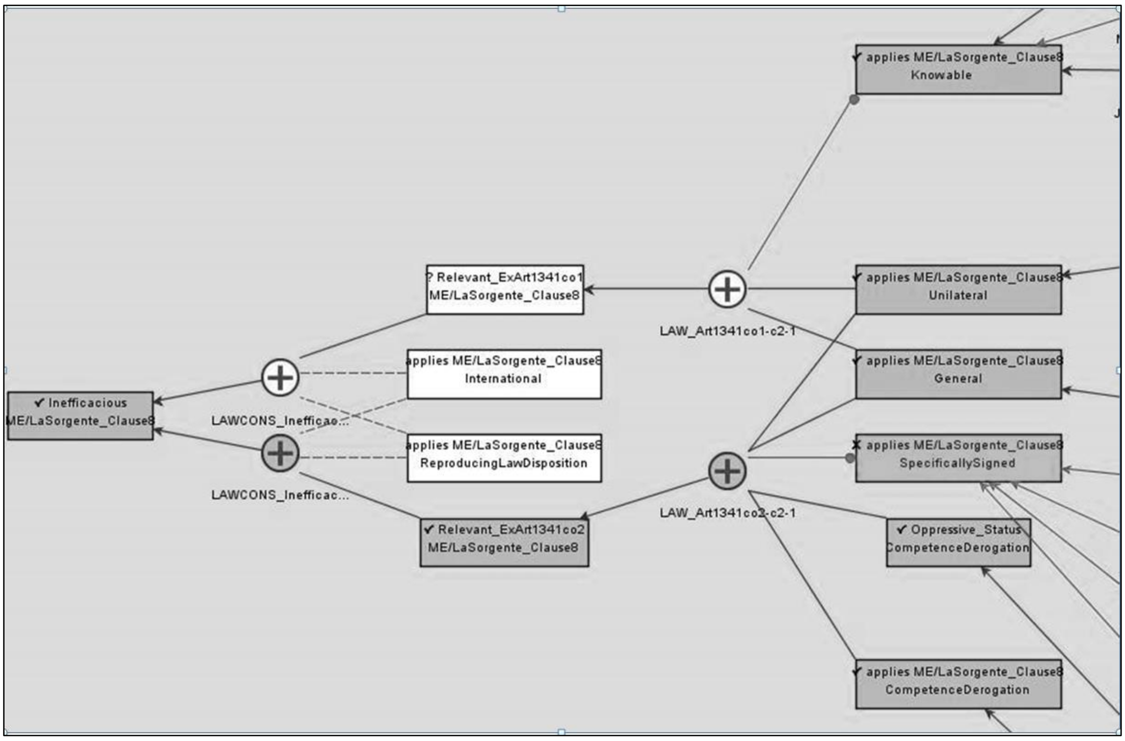


Fig. 4.4 – The graph visualizer.

The Carneades software includes a library for generating diagrams of argument graphs using Graphviz [Ellson et. al. 2001], which can export graphs in various file formats such as PDF and SVG[22].

The main feature of Carneades' argument visualization (similarly to [Verheij 2005]) is that Carneades' diagrams are views onto a mathematical model of argument graphs which can be modified through the user interface, rather than simply a single, unmodifiable model without a mathematical foundation as in Araucaria (see 1.3.4.4.) and Rationale[23], representing Wigmore's

[22] http://www.w3.org/Graphics/SVG/.

[23] http://Rationale.austhink.com.

(see 1.3.3.2.), Beardsley's (see 1.3.3.3.), and Toulmin's (see 1.3.3.4.) diagrams. When reconstructing Wigmore charts in Carneades, the key idea is to represent the various types of

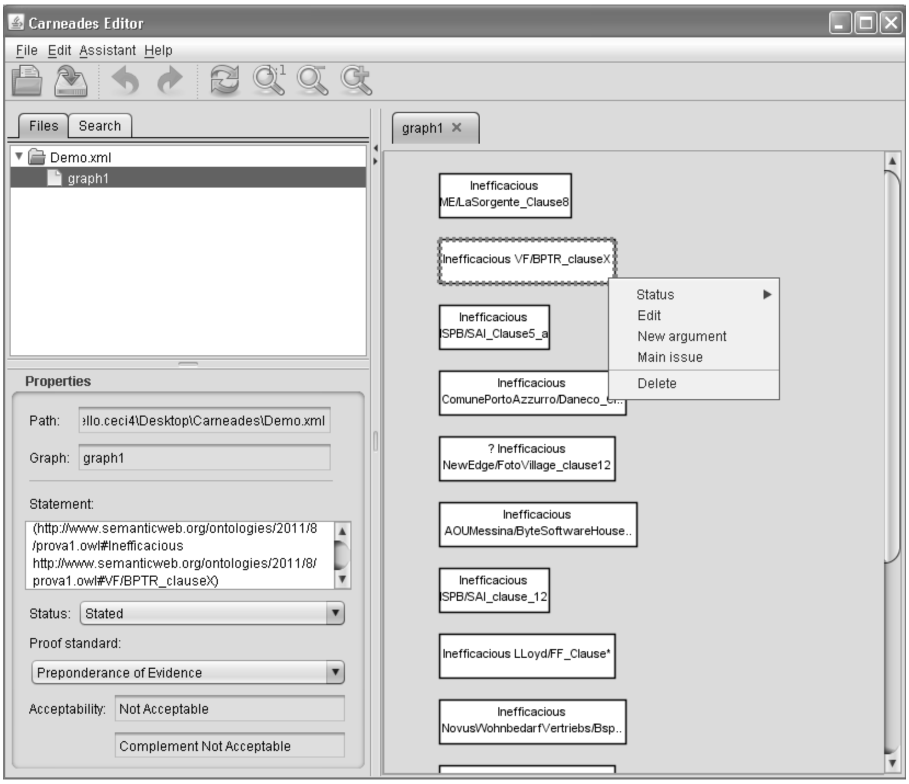


Fig. 4.5 – The GUI.

evidence with different argumentation schemes. Testimonial evidence, for example, can be understood as instantiation of a scheme for arguments from testimonial evidence [Gordon 2007b].

The distinction between the formal model and its visualization is very important to overcome typical diagramming issues which rise when roles and interactions in argumentation can be visualized in different ways. An example is the visualization of undercutters, arguments directly attacking the inferential link between the premises of an argument and its conclusion. In Carneades they are modeled as attacks on the major premise of an arguments, but other diagramming tools such as ArguMed [Verheij 2005] use a different method, called "entanglement", to visualize undercutters: there, the undercutting argument points to the arrow between the premises and conclusion of the argument being undercut. It is possible, however, to use this diagramming technique in the Carneades argument visualization, by modifying the way the concept of undercutter is translated in the graph.

4.2.5. Carneades 1.0.2

The theories and methodologies presented above were assembled in an application capable of graphically representing argumentation processes. The program, running on a Java Virtual Machine, is a work environment including a GUI and a graph visualizer which automatically represents argument graphs through boxes (for statements), circles (for arguments) and arrows (for the identification of the role of premises/assumptions/exceptions/conclusion of statements in relation to arguments).

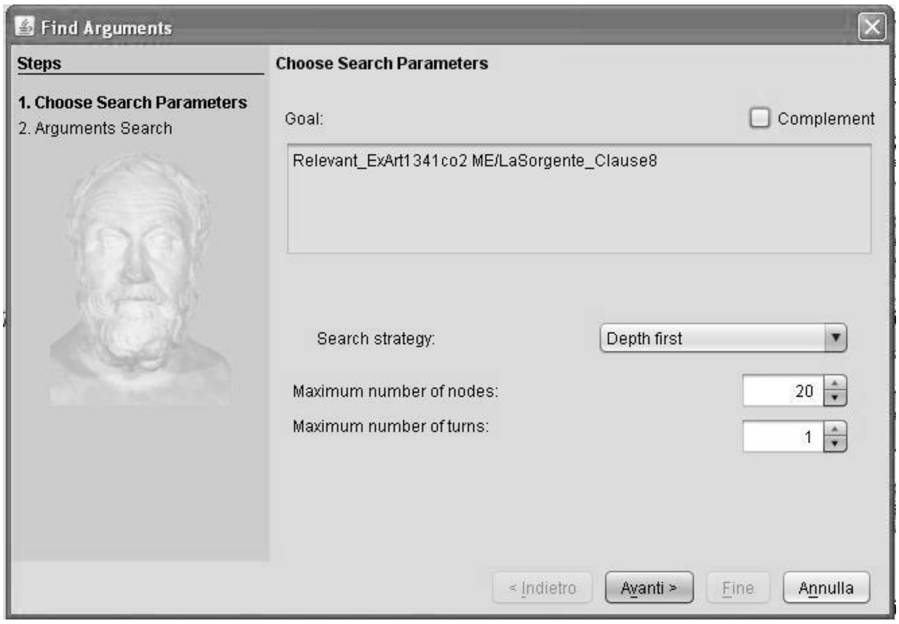


Fig. 4.6 – The *find argument* tool.

New premises and arguments can be manually added to the graph by using the specific tools located on the left hand side of the GUI. It is possible to specify the default status of the statement between stated, questioned, accepted, rejected. Arguments can be created and its premises/conclusion can be linked directly from the visualization window.

The *find argument...* tool exploits the Carneades reasoner to automatically find arguments pro and con the selected statement, marked as "target" for the argument search.

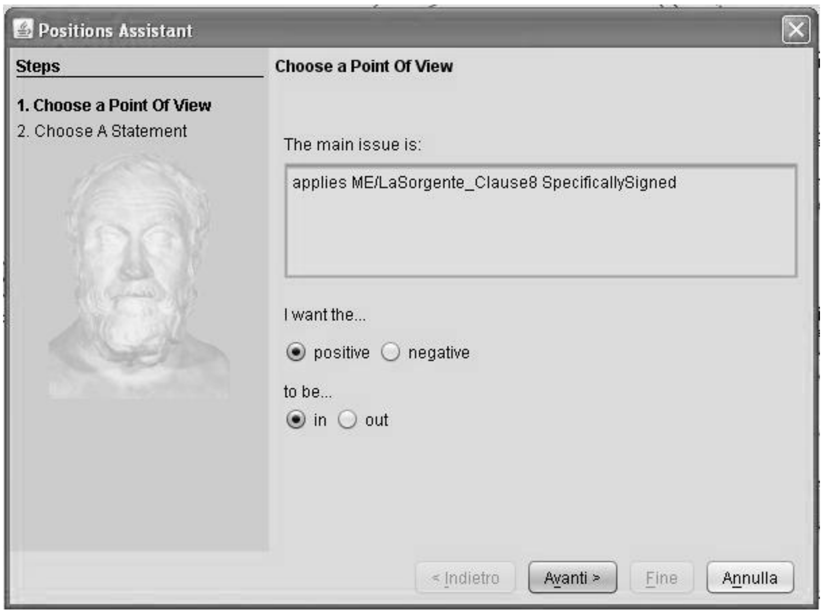


Fig. 4.7 – The *find positions* tool.

Another tool, called *find positions* and introduced in [Ballnat and Gordon 2010], provides suggestions about how to obtain the desired status for a target statement. In this case, Carneades will look for possible configurations of statements status which allow the target statement to be proved as requested, and feed them back to the user in an ordered fashion, giving precedence to positions which require the least number of questioned statements to be proved.

This reasoning can be conducted not only among the arguments which are already present in the graph, but can also rely on an automatic argument construction. This can be done by combining the *find argument* and *find positions* tools, although this is not an automatic function. In order for Carneades to automatically build arguments which bring to the acceptation/rejection of the target statement, its knowledge base must be enriched by importing a set of rules, representing templates of possible arguments, and a set of triples, representing accepted/rejected statements. While rules are imported as an XML file in LKIF-Rule language, the triples can either be imported from XML triples or in the form of OWL ontologies, whose not only explicit knowledge is imported, but also triples inferred using an in-built adaptation of the Pellet reasoner.

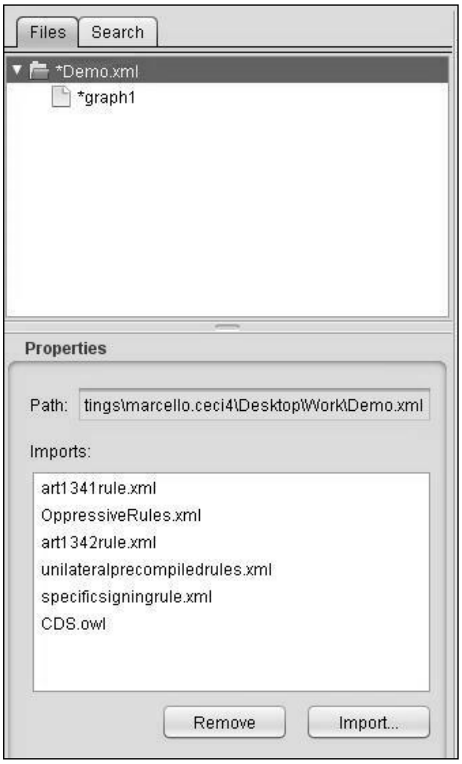


Fig. 4.8 – The import window.

4.2.6. Carneades 2.x

At the time of the research, Carneades is undergoing a vast renewal under different aspects, towards a new version which is going to be released in the near future.

An important change concerns the language used for rules representation: XML will be replaced by s-expressions. The objective is to improve readability computability and enhance a layered structure for expressing the rules' syntax, although Clojure (the language used for introducing s-expressions) presents some issues mainly due to its relying on Horn clause logics (see below 4.3.2.).

4.2.7. New Features

Following the indications coming from the testing of the first Carneades, the new semantics for assumptions and exceptions will be slightly modified. In Carneades 1.0.2, the qualification of the premise type does not influence the statement itself, but rather the link between the statement and the argument. This solution had the disadvantage of allowing incoherent behaviours of statements, which resulted to be false and true at the same time[24]. Therefore,

[24] in Carneades 1.0.2, if a statement appears as the assumptions for one argument, and the exception to one other argument, and if it appears as stated, this would not prevent neither of the two arguments from being valid. Hence, the reasoning on the argument graph would stem from the (assumed) acceptability of the statement on one side (assumption) and on the (assumed) rejection of the same statement on the other side (exception).

Carneades 2.x presents a new modelling based on default weights which are assigned to statements, ranging on a scale from 0 to 1 as follows:

0.00 rejected

0.25 assumed false

0.50 stated

0.75 assumed true

1.00 accepted

The GUI has been redesigned into a multiuser, three-tiered Web-based interface. This change in the approach to the system is coherent not only with the recent trends in academic and experimental programming, but also with Carneades' attitude to enhance semantic interaction in a social environment. A web-based interface would allow a more fluid integration of Carneades with online repositories and sources for knowledge, rules, and argumentation schemes. It could

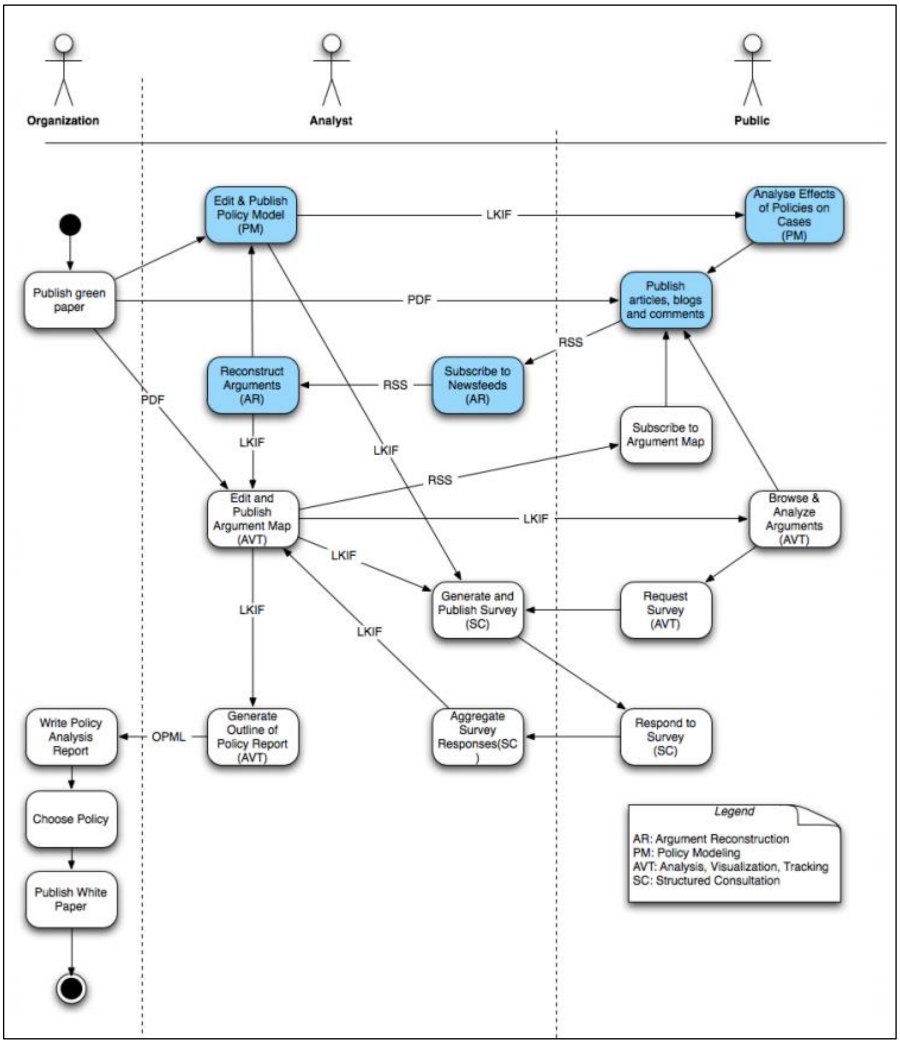


Fig. 4.9 – Process Model of the Policy Modelling Tool in [IMPACT 2013]

include a public repository of argument graphs, which would strongly enhance the comprehension of this system's potentialities.

4.2.7.1. The Policy Modeling tool

The new version of Carneades is implemented into a wider Policy Modeling tool. The tool uses computational models of policies, applying methods from AI & Law and Computational Models of Argument, to help users understand the legal effects of alternative policies in particular fact situations or cases. The Tool's applications are those eGovernment, eDemocracy-oriented applications which aim at helping citizens and other stakeholders to better understand the proposed policies, so as to better enable them to contribute informed arguments to policy debates Inside this suite, the goal of the Carneades Argumentation System is to represent argumentation theory of philosophy into a computational tool through formal model. This model is exploited by several tools which fulfill different argumentation-related tasks such as:

- Argument reconstruction;
- Argument visualization and browsing;
- Structured surveys, similar to Parmenides [Atkinson 2006];
- Policy analysis, realizing a kind of rule-based expert system.

The system presents itself as a multiuser, web-based service. Similarly to version 1.0.2 it runs on Java Virtual Machine, which ensures interoperability, and brings with him its web server and database engine. Doesn´t need to be configured and is currently available in open-source format at carneades.github.com.

As explained in Figure 4.9, the tasks fulfilled by the tools involve different actors, which access – as users – the same tools for different purposes: starting from the right: the authority to present issues for the debate, participants to answer polls and navigate argument graphs and the moderator to model the argument templates in order to extract useful data from these polls.

The tasks that can be carried out by the specific tools are:

- **Summarizing** the arguments of a debate in an argument graph such as that of Figure 4.10;
- **Visualizing**, **browsing** and **navigating** argument graphs;
- Critically **evaluating** arguments;
- Obtaining clear **explanations**, using argument graphs, of the different effects of alternative policies in particular cases;
- Forming **opinions**, participating in **polls** and **ranking** stakeholders by the degree to which they share your views.

At the current state, four tools of the suite are already implemented: the argument formalization, the graphs, the polling system and the policy analysis tool. A manual is available in draft version, and the whole system is packaged in a self-installing java file.

Possible applications of the Policy Modelling Suite include:

- **eDemocracy/eParticipation**, improving the quality, efficiency, inclusiveness and transparency of democratic public policy deliberations;
- **Claims processing**, improving the efficiency and correctness of claims processing procedures, both in public administration and the private sector, for example with regard to social benefits or insurance claims;
- **Regulatory compliance**, helping companies to comply with regulations;
- **Case management**, better managing complex legal cases in law firms, by creating maps of relationships between claims, legal arguments, case and statute citations, and testimonial and documentary evidence;
- **Humanities education**, facilitating the learning of critical thinking and analysis skills, particularly in law, philosophy, religion and other humanities fields, where competing theories are constructed by interpreting texts.

4.3. The modelling of judicial precedents with Carneades

4.3.1. Version 1.0.2

The Carneades argumentation system and its background theory were used in the present research to enrich the semantics of the Legal Ontology beyond the mere classification of claims,

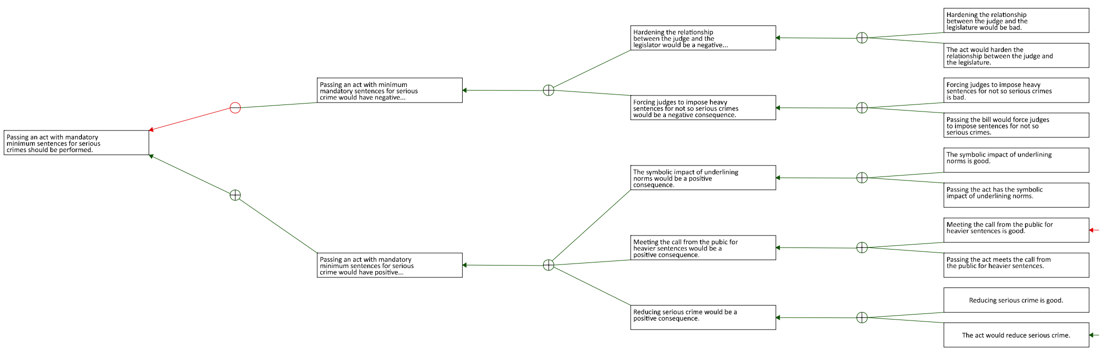


Fig. 4.10 – An example argument graph in Carneades 2.x

legal statuses and judicial interpretations on the basis of the considers/applies properties: the ontology set is in fact capable of telling which concepts are used in a decision and how they are used, but it cannot tell why (see Chapter 3). In fact, it is impossible to infer the so-called *ratio decidendi* followed by the judge from the mere binding of a material circumstance to a legal status. Moreover, as seen in 3.5.2., DL axioms allow defining the concept of *applicable rule* (or, from the opposite perspective, *relevancy under article x*) and to infer knowledge about the applicability of each rule on each case, but only to a limited extent. In fact, exceptions (as well as assumptions) can be represented, but will not bring any inferential capability because of the open world assumption. Moreover, the insertion of an exception into the axiom of an “applicable” class will in fact prevent any inference on the members of that class unless the declared knowledge base is complete (i.e. by explicitly stating which statuses do not belong to a particular individual, which means explicitly stating, for each status which is used as an exception in the axiom, not only the individuals who are actually bound to it but also those who aren’t). This brings down much of the power of the system built so far, and therefore the ontology set needs to be attached to some extension which could handle defeasible logics, in order to allow more complex reasoning on the knowledge contained in the decisions’ texts.

The aim of the research done on the Carneades system is to create a suitable environment where the knowledge extracted from the decision’s text can be processed and managed, in such a way as to enable a deeper reasoning on the interpretation instances grounding the judicial decision. Examples of this deeper reasoning include:

- **Finding** relevant **precedents** which were not explicitly cited in the decision;
- **Finding anomalies** in the evaluation of material circumstances, in the light of cited precedents and similar cases;
- **Validating** the adjudication(s) of the judge on the claim(s) brought forward by the parties during the trial, on the basis of applicable rules, accepted evidence and interpretation;
- **Suggesting** possible **weak spots** in the decision’s groundings;
- **Suggesting** possible **appeal grounds** and legal rules/precedents/circumstances that could bring to a different application of the rules and/or to a different adjudication on the claim.

This is possible thanks to the mix of OWL-DL reasoning, semantically managing static information on the elements of the case, and rule-based defeasible reasoning, which ought to represent the dynamics of norms and judicial interpretations. The present approach focuses on the *argument from ontology* feature of Carneades: the program is in fact capable of accepting (or rejecting) the premises of arguments on the basis of the knowledge contained in some imported OWL/RDF ontology (see below 3.1). This allows to build complex argumentation graphs, where the argument nodes represent legal rules and the statements are accepted or rejected also on the basis of knowledge coming from the ontology and/or data inserted by the user.

In this perspective, the Carneades argument graph may either represent:

- A **reconstruction of a judicial decision's contents** in terms of laws applied, factors taken into considerations, and interpretations performed by the judge. The conclusion of the argumentation represents the final adjudication of the claim, and the Carneades reasoner will accept or reject the claim as a result of the application of the judicial interpretations contained in the decision's groundings (this is the kind of representation which will be shown in the present application);

- A **collection of argumentations paths** leading to a given legal statement (such as "contract x is inefficacious"). On the basis of manually-inserted statements concerning the object of the case (statuses or factors concerning the material circumstance, i.e. contract x) the Carneades reasoner suggests possible argumentation paths leading to the acceptation (or rejection) of the desired legal statement.

In both cases, however, the system presents to the user not only argumentation paths which have been proved as valid (i.e. rules whose conditions have all been met), but also possible, incomplete argumentation paths where one or more of the premises is still undecided: under this perspective, Carneades provides a semantic environment where different laws, legal statuses and precedents are related to each other. From that point, the user can go further by querying the knowledge base to retrieve precedents where similar (or different) interpretations are made: in this way, he can realize which differences – if any – exist between two or more precedents. It is like browsing case-law in a law journal in order to compare different decisions, but in the Carneades environment this can be done directly with legal concepts, not only to verify a combination of circumstances and laws under a logical point of view, but also to receive suggestions from the system on which law, precedent or circumstance could lead to a different outcome.

The implementation of Carneades has been carried through different tasks:

- **Enriching the semantic content** of the Legal Ontology by representing the fine-grained knowledge contained in the decision’s text, in an environment where this expansion of the knowledge base does not entail an overburdening of the OWL reasoners (which would compromise the usability of the tool due to the exponential growth of the reasoning time when enlarging ontology-based KBs);

- **Modelling a rule system** representing the dynamic relationships created by judicial interpretations and law applications, providing the basis for answering a wide range of queries about the case in the light of (cited and uncited) precedents;

- **Importing the knowledge of the ontology** set in such a way that the two can successfully interact, allowing exchange of knowledge from the Ontology Set to the Carneades model (and, possibly, also vice versa).

4.3.2. Ontology Import

The first activity consisted in verifying the import system in Carneades 1.0.2, which resulted to be complete (importing both declared and inferred knowledge). In this initial phase, the Legal Ontology imported is still the complete ontology set (core+domain) containing axioms for `Relevant` clauses as well as `Inefficacious` clauses. The consistency of the imported knowledge was verified through simple queries, asking Carneades about the acceptability of some assertion, using only the knowledge base imported from the ontologies. In these simple cases, the representation of the argumentation scheme was rather simple and brought no new information on the surface. As it can be seen in Figure 4.11, the only arguments coming into play are *arguments from ontology*. The argument named HermiT represents the inference made by the reasoner, and the premise "valid http://www.semanticweb.org/o.." is the (assumed) premise that the ontology at that URL is valid. This behaviour is related to the import system used in Carneades 1.0.2: all knowledge (explicit and derived) is imported without keeping track of their origin. Therefore, it is not possible to argue about the inferences made by the ontology reasoner, or to analyze/modify/exclude any of the axioms bringing inferences. Moreover, it is not possible to use these axioms to process new information coming out from Carneades' argumentation engine: all statements which do not come from the knowledge base of the ontology, but rather use external knowledge (Carneades rules or custom statements) will not be processable under the ontology's axioms, and vice versa. This problem in re-feeding new knowledge on the ontology inferencing engine is a major issue, which will be further discussed later in 4.5.3.

4.3.2.1. Adding Factors

The next step is to enrich the knowledge base with information coming from the decision's text, where not only the high legal concepts (concepts directly coming from the law text) were used, but also lower, more blurry concepts (called *factors*)

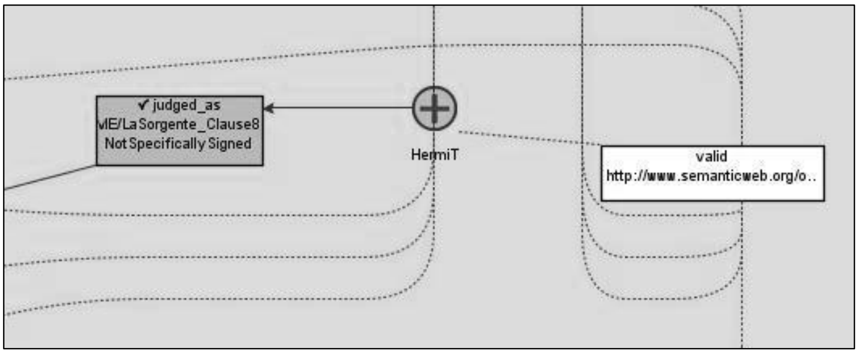


Fig. 4.11 – Example of argument from ontology in the argument graph.

come into play. It is important to make a distinction between *legal statuses* and *factors*, but at the same time the border between these two concepts is not very easy to identify. For example, the legal status `AddedToPrecompiled` (stating that a specific clause has been manually added to a precompiled contract) appears as part of a main premise in rule Article 1342co1, but also constitutes an important factor in judicial interpretations towards the attribution of the `Knowable` legal status (relevant in rule Article 1341co1). The circumstance of the clause being added to a precompiled contract can therefore be represented as a legal status, as a factor, or as both of them. Nevertheless, it is necessary to either define a standard in distinguishing between these two, or to renounce to this distinction entirely, using only one class and a couple of properties (this issue on properties, trying to set a flexible and modular ratio between consider/applies, hasfactor and judged_as will be further discussed in 4.5.7.). The present research maintains an open approach in defining those concepts, allowing the markup of an element as both *factor* and *legal status*.

Even if they could semantically be treated equally, the syntactic involvements of factors are much looser than those of legal statuses: once a legal status is assigned to a material circumstance (either through explicit insertion, interpretation or reasoning), a series of possible consequences may be triggered, including statements concerning law applicability or legal consequences. On the other hand, when a factor is `applied_by` a material circumstance, the possible consequences include only the possible assignment of a legal status, but this is never

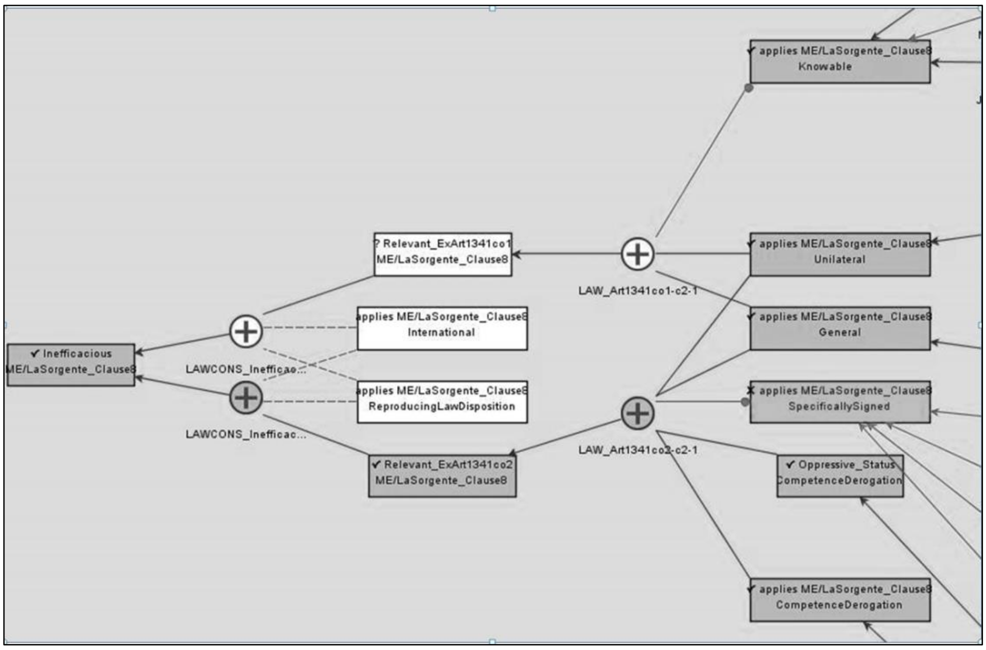
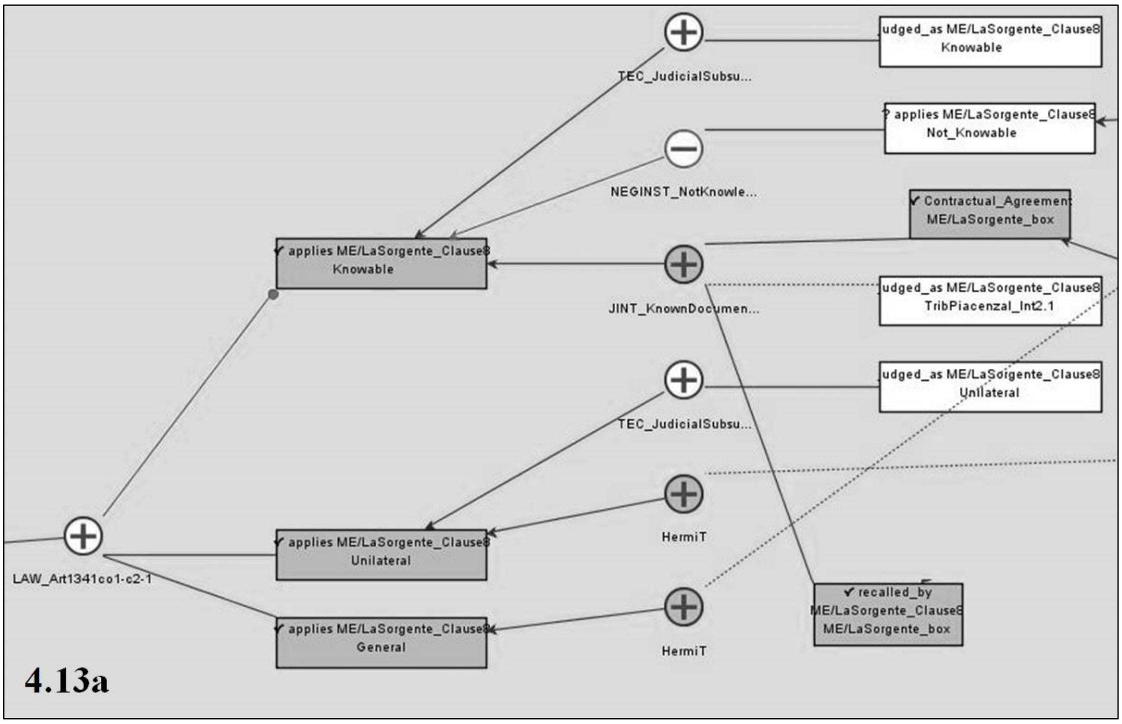
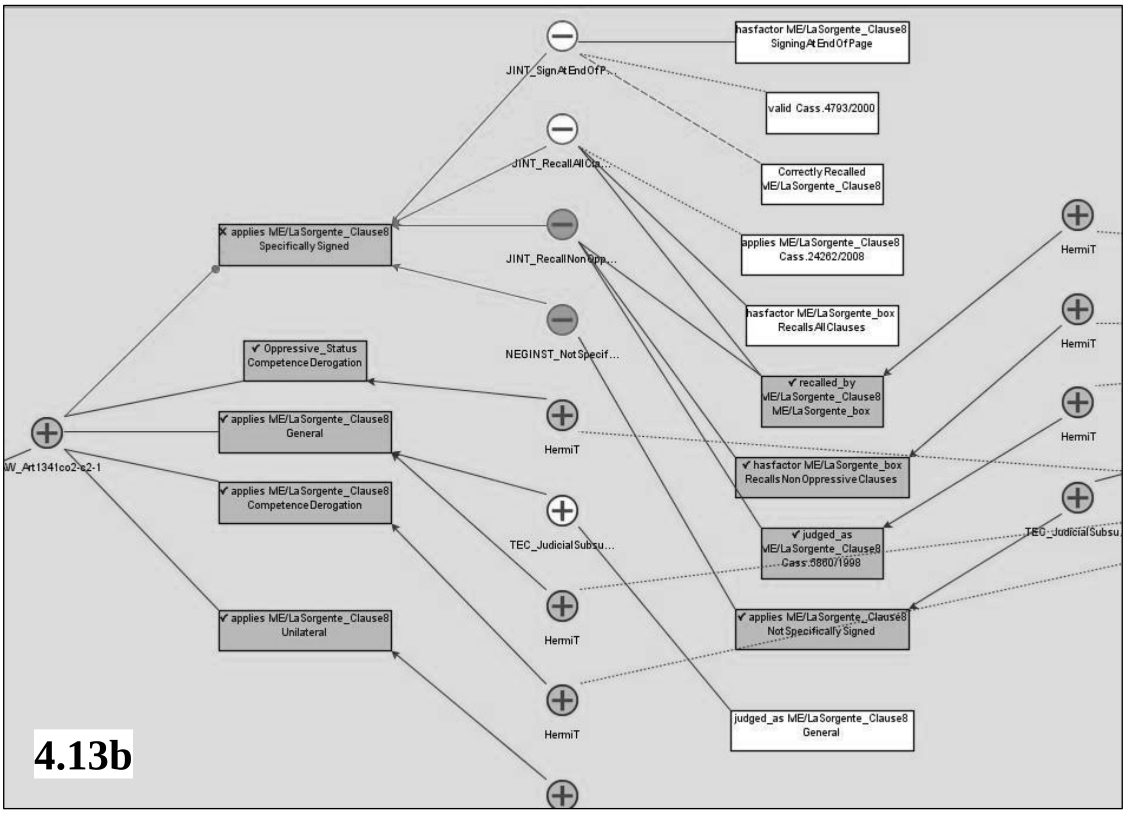


Fig. 4.12 - Example of legal status(es) to rule application.


judged_as ME/LaSorgente_Clause8 Knowable
TEC_JudicialSubsu...
? applies ME/LaSorgente_Clause8 Not_Knowable
NEGINST_NotKnowle...
✔ Contractual_Agreement ME/LaSorgente_box
✔ applies ME/LaSorgente_Clause8 Knowable
judged_as ME/LaSorgente_Clause8 TribPiacenzal_Int2.1
JINT_KnownDocumen...
judged_as ME/LaSorgente_Clause8 Unilateral
TEC_JudicialSubsu...
HermiT
✔ applies ME/LaSorgente_Clause8 Unilateral
LAW_Art1341co1-c2-1
✔ recalled_by ME/LaSorgente_Clause8 ME/LaSorgente_box
✔ applies ME/LaSorgente_Clause8 General
HermiT

4.13a


hasfactor ME/LaSorgente_Clause8 SigningAtEndOfPage
JINT_SignAtEndOfP...
valid Cass.4793/2000
Correctly Recalled ME/LaSorgente_Clause8
JINT_RecallAllCla...
✘ applies ME/LaSorgente_Clause8 Specifically Signed
JINT_RecallNonOpp...
applies ME/LaSorgente_Clause8 Cass.24262/2008
HermiT
hasfactor ME/LaSorgente_box RecallsAllClauses
NEGINST_NotSpecif...
✔ Oppressive_Status Competence Derogation
✔ recalled_by ME/LaSorgente_Clause8 ME/LaSorgente_box
✔ applies ME/LaSorgente_Clause8 General
W_Art1341co2-c2-1
✔ hasfactor ME/LaSorgente_box RecallsNonOppressiveClauses
✔ applies ME/LaSorgente_Clause8 Competence Derogation
TEC_JudicialSubsu...
✔ judged_as ME/LaSorgente_Clause8 Cass.5860/1998
✔ applies ME/LaSorgente_Clause8 Not Specifically Signed
✔ applies ME/LaSorgente_Clause8 Unilateral
judged_as ME/LaSorgente_Clause8 General

4.13b

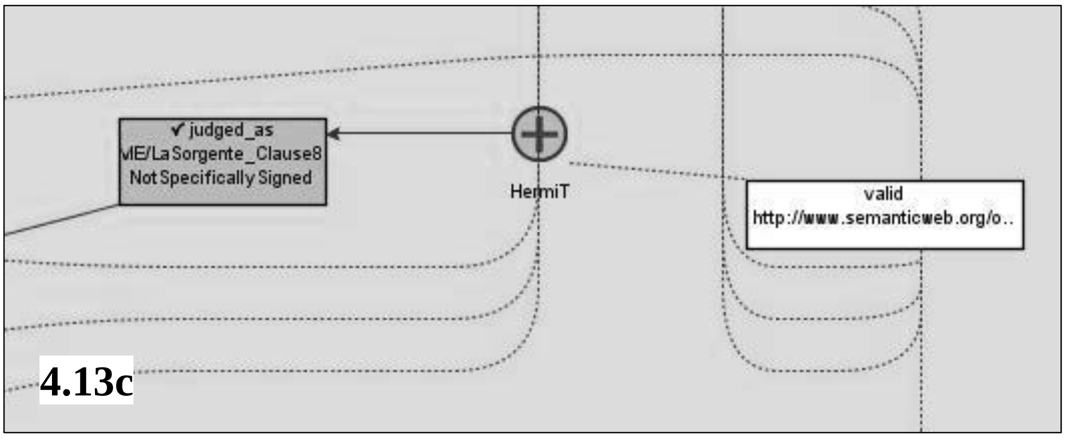


Fig. 4.13 (a, b, c) – Example of factors to legal statuses.

automatic: the argument bringing to the assertion "`applies A X`" has always some assumed premise in addition to the factor(s) required for that interpretation. The assumption consists in some judgement or doctrine to be accepted in the specific case. This, of course, does not happen when applying legal rules, which do not require acceptance (being rules subject to different exceptions, e.g. those coming from the rules hierarchy).

The link between the factor and the consequent status is therefore weak and defeasible. On the other hand, concepts expressed by the factors are broad and can be shaped differently when they appear in different precedents. In this way, when reasoning about factors, the system is capable of bringing forward many possible interpretations, assigning different combination of factors to the material circumstance, thus bringing a wide and hopefully inspiring list of possible interpretations of the fact. These suggestions are also ranked by Carneades through the *find positions* tool (Figure 4.7), which suggests first the argumentation which require less yet unknown information to be asserted, while at the same time using many already accepted (or rejected) assertions as valid premises.

The following list describes the factors individuated in the decisions sample, divided into categories which depend on the legal status(es) whose interpretation the factors lead towards. What follows is indeed a partial, open list.

Factors related to a contract clause's oppressiveness (legal status: Oppressive_Clause):

– **InsuranceCoverageLimitation**: the clause contains a limitation in the coverage of the insurance contract;

– **RiskExclusionForElectricPhenomena**: the clause contains an exclusion of the insurance coverage for damages consequent to electric phenomena;

– **RiskExclusionForElectricDevices**: the clause contains an exclusion of the insurance coverage for damages occurred to electric devices;

– **InsuranceRiskExclusion**: the clause significantly reduces the insurance coverage of the risk;

- **InsuranceObjectSpecification**: the clause defines the object of the insurance coverage;
- **InsuranceDrunkDriverExclusion**: the clause excludes the insurance coverage for damages caused by a drunk driver;
- **SoleRightInAgency**: the clause states that the agent has to be the sole agent for the counterpart (in a territory);
- **WithdrawalForBothParties**: the clause allows both parties to withdraw from the contract;
- **ExpertOpinion**: the clause forces the parties to rely on an expert's opinion in case of disputes on the contract's content;

Factors related to a contract unilaterality or precompliedness (legal statuses: Unilateral; Precompiled):

- **Precompiled**: contract was precompiled (previously prepared by one of the parties);
- **GeneralClauseSubsumption**: bla;
- **Peradesione**: the contract was of a particular tipe, called *per adesione* in Italian Civil Code: it is a kind of contract where the acceptant can only accept the fixed conditions set by the proponent;
- **Negotiated**: the contract has been object of negotiations between the parties.;
- **ObjectExcludesPeradesione**: the object of the contract excludes it being a contract *per adesione*;

Factors related to a clause's specific signing (legal status: SpecificallySigned):

- **RecallObjectOrNumber**: the space devoted to specific signing recalls the object or the number of the clause;
- **RecallObjectAndNumber**: the space devoted to specific signing recalls the object and the number of the clause;
- **RequiredObjectOrNumber**: the present judgement considers sufficient the indication of object or number of the clause;
- **RequiredObjectAndNumber**: the present judgement requires the indication of both object and number of the clause;
- **RecallAllClauses**: the space devoted to specific signing contains a recall of all the clauses of the contract;
- **RecallNonOppressiveClauses**: the space devoted to specific signing contains a recall of clauses of the contract which are not oppressive;

– **SpecificSigningThroughRecall**: the clause is specifically signed through a signing in a dedicated space distinct from the clause itself;

– **SingAtEndOfPage**: the page containing the clause has been signed;

– **ProductionDuringTrial**: the clause has been produced during the trial;

– **KnownDocument**: the document is known to the parties;

– **KnownDocumentRecall**: the clause consists in a document recalled by the contract;

– **InsuranceDocumentRecall**: the clause consists in an insurance document recalled by the contract.

4.3.3. Creating rules

In order to support automatic argument construction within Carneades 1.0.2, legal rules (those formerly included in the Legal Ontology as "axioms") and interpretation instances have been represented in LKIF-Rules language. It is possible to distinguish three kinds of rules used to support automatic argument construction in the present research:

Rules representing law relevancy: they bring together different legal statuses and apply a "relevance status" to the material circumstance which considers these legal statuses. Here is an example:

```
<rule id="LAW_Art1341co1">

    <head>

        <s pred="Relevant_ExArt1341co1"><v>C1</v> falls under
        the discipline of Article 1341 comma 1 of Civil Code
        </s>

        <s pred="&oss;considered_by"><v>C1</v> falls under the
        discipline of <i value="&oss;Art1341co1cc">Article
        1341 comma 1 of Civil Code </i></s>

    </head>

    <body>

        <s pred="&oss;applies"><v>C1</v> applies <i
        value="&oss;General"> general status</i> </s>

        <s pred="&oss;applies"><v>C1</v> applies <i
        value="&oss;Unilateral"> unilateral status</i> </s>

        <not><s pred="&oss;applies"><v>C1</v> not applies <i
        value="&oss;Knowable"> knowable before contract
        sign</i> </s></not>

    </body>

</rule>
```

Rules representing law application: the application of a legal consequence is a further step, and is represented by a different rule. This is the phase when possible exceptions to the application of the rule may come into play. Here is an example:

```
<rule id="LAWCONS_Inefficacy rule">

    <head>

        <s pred="&oss;Inefficacious"><v>C1</v> Is
        inefficacious: has no effects </s>

    </head>

    <body>

        <and>

            <or>

                <s pred="Relevant_ExArt1342co2"><v>C1</v>
                falls under the discipline of Article
                1342 comma 2 of Civil Code </s>

                <not exception="true"><s
              pred="&oss;applies"><v>C1</v> applies <i
              value="&oss;ReproducingLawDisposition"> a
              law disposition</i> </s></not>

                <not exception="true"><s
        pred="&oss;applies"><v>C1</v> applies <i
        value="&oss;International"> an international
        agreement</i> </s></not>

            </or>

        </and>

    </body>

</rule>
```

Rules representing precedents: these can be called *rules of precedent*, because they actually bring together different factors and apply a legal status to the material circumstance which considers these legal statuses. Since the actual applicability of the precedent is not a matter of logic defeasibility, but rather depends on the evaluation of the judge, this rule only brings forward an assumption that the material circumstance could be interrpeted in the light of a certain precedent. Therefore, every *rule of precedent* includes assumptions which represent the acceptance of the precedent, and shares its name. Here is an example:

```
<rule id="JINT_ProducedDuringTrial">

    <head>
```

```
<s pred="&oss;applies"><v>C1</v> applies <i value="&oss;SpecificallySigned"> specifically signed status </i></s>

</head>

<body>

<s pred="&oss;hasfactor"><v>C1</v> has factor <i value="&oss;ProducedDuringTrial"> is produced during trial</i></s>

<assumable="true"><s pred="&ossjudged_as"><v>C1</v> is judged as with <i value="&oss;Cass.460/1983">precedent Cass. 460/1983</i> is accepted</s>

<not exception="true"><s pred="&osshasfactor"><v>C1</v> has factor<i value="&oss;ProducedNotToEnact"> was produced with the intention non to enact it</i></s></not>

</body>

</rule>
```

Rules enhancing semantics: managing statuses and factors in a specific domain, it is necessary to introduce some semantic rules that allow some more inference on the possible status and factor applicable to a specific circumstance. These rules do not directly come from law, and neither from judicial interpretation. Their role is to establish a link between low- and high-level factors, so that more complex concepts can be represented with a single factor in higher rules, and at the same time the semantics of these complex factors can be kept intact, allowing the factor to be bound to material circumstances which have the required characteristics, but expressed in lower-level concepts. These rules may come from taxonomies, contract dispositions, natural or scientific evidences, or even very strong legal doctrine. It doesn't comprehend exceptions (as in legal rules) or assumptions (as in *rules of precedent*) and therefore does not constitute a "node" in the defeasible argumentation process. Nevertheless, it is necessary to represent this "taxonomy" in LKIF-Rule because of the problem with importing the ontology and its axioms: if represented only in OWL, in fact, Carneades would only be able to apply the axioms to the (stated and inferred) knowledge of the Ontology Set, and would be unable to repeat the reasoning on new knowledge coming out of the application of the LKIF rules. If the solution of translating axioms into rules is adopted (see the issue in 3.4.3.3), this whole category could be moved to the Ontology Set (which is its preferrable place, being a "taxonomic" classification rather than a defeasible logics language). Here are two examples:

```
<rule id="TEC_RecallObjectORNumber">

<head>

<s pred="Recalls_ObjectORNumber"><v>B1</v> recalls object or number</s>

</head>
```

```
        <body>

                <or>

                     <s pred="&oss;hasfactor"><v>B1</v> has factor
                     <i value="&oss;RecallsNumber"> recalls
                     number</i> </s>

                     <s pred="&oss;hasfactor"><v>B1</v> has factor
                     <i value="&oss;RecallsObject"> recalls
                     object</i> </s>

                </or>

        </body>

</rule>

<rule id="NEGINST_NotGeneral">

        <head>

              <not><s pred="&oss;applies"><v>C1</v> applies <i
              value="&oss;General"> general status </i></s></not>

        </head>

        <body>

              <s pred="&oss;applies"><v>C1</v> applies <i
              value="&oss;Specific"> specific status </i></s>

        </body>

</rule>
```

Under a different perspective, the first three types of rules represent three phases of a procedure, which is anyway not to be followed strictly:

– **Rules representing precedents** take into account stated (and possible) factors to be applied to the various material facts of the case, highlighting possible precedents that apply some legal status to the circumstances;

– **Rules representing law relevancy** take into account stated (and possible) legal statuses to be applied to the various material circumstances of the case, highlighting relevance under legal rules. The system distinguishes between legal statuses contained (or inferred) in the knowledge base of the Ontology set, and those coming from rules representing precedents;

– **Rules representing law application** finally take into account circumstances which are already relevant to some law, verifying that no exceptions (i.e. coming from other laws) apply to the case, and if none applies the circumstance is labeled with the legal consequence provided by the relevant law.

This proceeding can also be tracked backwards, starting from the simple query about some specific legal consequence being applicable to a material circumstance (i.e. a contract). The system then builds a tree of possible argumentation paths, including all the four types of rules presented above. Any of the statements introduced by the reasoner can be then further searched for arguments: if a law relevancy statement is highlighted, rules representing law relevancy will be used to show plausible paths that could lead to the desired consequence (accept/reject the statement), while if a statement applying a legal status to a circumstance is highlighted, rules representing precedent will be used to show precedents that lead to the desired consequence.

4.3.4. The query graph

In order to show the argument building process undergone by Carneades, a *demo.xml* Carneades argument graph was created. The graph includes 27 issues, one for each sentence of the research sample, imported from the OWL judicial ontology set together with several module containing the rules used in the sample sentences (see Appendix). Each issue concerns the application of a specific legal consequence (`Inefficacious`) to a specific material circumstance (i.e. `ME_LaSorgente_Clause8`). Through the *find argument* tool, Carneades builds the argumentation tree in Figure 4.14.

Asking Carneades to verify the acceptability of a query, the system assigns a *dialectical status* to the main issue, showing also the argumentative path followed to come to the conclusion.

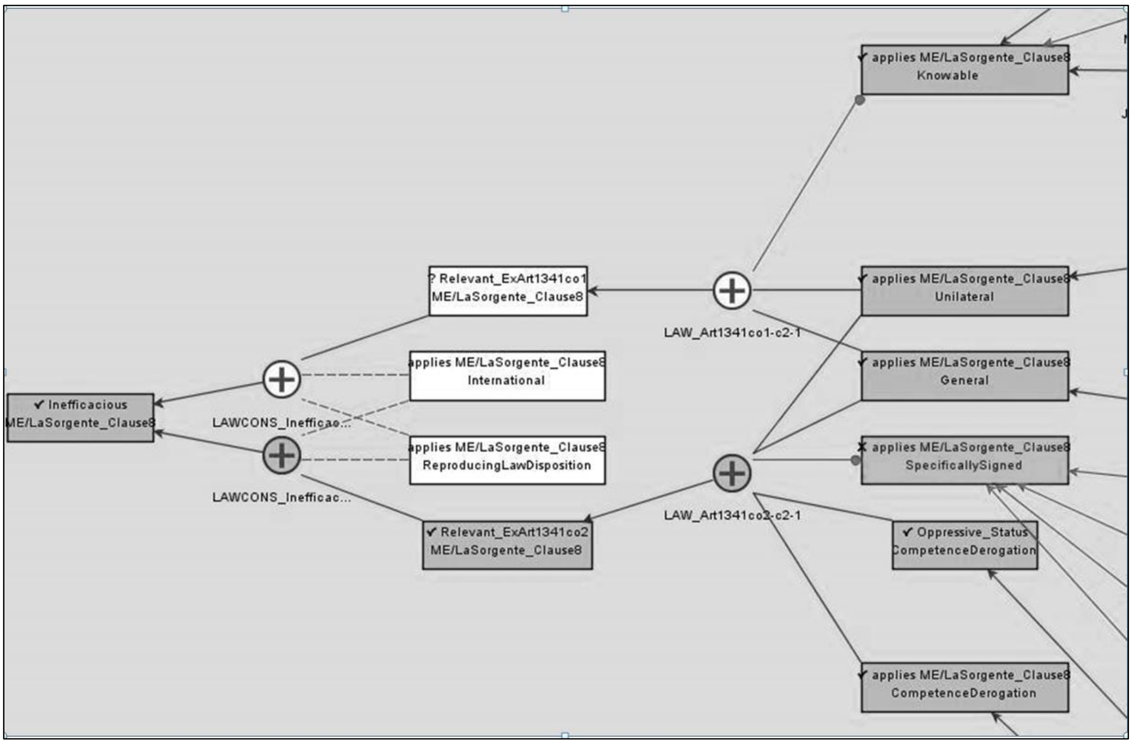


Fig. 4.14: example of Carneades argumentation tree on the decisions sample.

Highlighting further statements in the path, it is possible to ask Carneades to analyze the question in further details, suggesting possible logic and juridical paths through which it could be possible to come to the acceptance of (or rejection of, or to not decide) the highlighted statement. In this way, it is possible for the user to explore the interpretation instances which play some role in the decision, scouting possible paths that could lead to a different outcome for the claim.

4.3.5. Classification of Judgements through Adjudication

Carneades rules can be used to reproduce not only the reasoning of the judge when dealing with interpretations (linking a material circumstance to a legal status) but also when dealing with adjudication (linking a legal claim to a legal outcome). In particular, the following rules (in LKIF-Rule language) was conceived:

```
<rule id="Adjudication">
    <head>
        <s pred="&oss;accepts"><v>J1</v> accepts <v>C1</v></s>
    </head>
    <body>
        <s pred="&oss;Judgement"><v>J1</v> is a judgement </s>
        <s pred="&oss;applies"><v>J1</v> applies <v>S1</v></s>
        <s pred="&oss;applies"><v>S1</v> applied_by
<v>C1</v></s>
        <s pred="&oss;Claim"><v>C1</v> is a claim </s>
        <s pred="&oss;considered_by"><v>C1</v> is considered
        by <v>J1</v></s>
    </body>
</rule>
<rule id="Adjudication-plaintiff">
    <head>
        <s pred="&oss;ProPlaintiff"><v>J1</v> was decided pro-
        plaintiff </s>
    </head>
    <body>
        <s pred="&oss;Judgement"><v>J1</v> is a judgement </s>
        <s pred="&oss;accepts"><v>J1</v> accepts <v>C1</v></s>
        <s pred="&oss;held_by"><v>C1</v> is held by
<v>P1</v></s>
        <s pred="&oss;hasrole"><v>P1</v> has role <i
        value="&oss;Plaintiff"> plaintiff </i></s>
```

```
	</body>
</rule>
<rule id="Adjudication-defendant">
	<head>
		<s pred="&oss;ProDefendant"><v>J1</v> was decided pro-
		defendant </s>
	</head>
	<body>
		<s pred="&oss;Judgement"><v>J1</v> is a judgement </s>
		<s pred="&oss;accepts"><v>J1</v> accepts <v>C1</v></s>
		<s pred="&oss;held_by"><v>C1</v> is held by
<v>P1</v></s>
		<s pred="&oss;hasrole"><v>P1</v> has role <i
		value="&oss;Defendant"> defendant </i></s>
	</body>
</rule>
<rule id="Rejected-plaintiff">
	<head>
		<not><s pred="&oss;ProPlaintiff"><v>J1</v> was not
		decided following plaintiff's claim </s></not>
	</head>
	<body>
		<s pred="&oss;Judgement"><v>J1</v> is a judgement </s>
		<not><s pred="&oss;accepts"><v>J1</v> accepts
		<v>C1</v></s></not>
		<s pred="&oss;held_by"><v>C1</v> is held by
<v>P1</v></s>
		<s pred="&oss;hasrole"><v>P1</v> has role <i
		value="&oss;Plaintiff"> plaintiff </i></s>
	</body>
</rule>
<rule id="Rejected-defendant">
```

```
        <head>

              <not><s   pred="&oss;ProDefendant"><v>J1</v>   was   not
              decided following defendant’s claim </s></not>

        </head>

        <body>

              <s pred="&oss;Judgement"><v>J1</v> is a judgement </s>

              <not><s      pred="&oss;accepts"><v>J1</v>      accepts
              <v>C1</v></s></not>

              <s    pred="&oss;held_by"><v>C1</v>    is    held    by
  <v>P1</v></s>

              <s    pred="&oss;hasrole"><v>P1</v>    has    role    <i
              value="&oss;Defendant"> defendant </i></s>

        </body>

  </rule>
```

This allows the reasoner to classify the claim following the actual judge’s decision, distinguishing the following cases:

- If it applies the same rules of one of the claims, that claim is **accepted**;
- If it doesn’t, that claim is **rejected**;
- If a “third solution” is found by the judge, **both** claims appear as **rejected**.

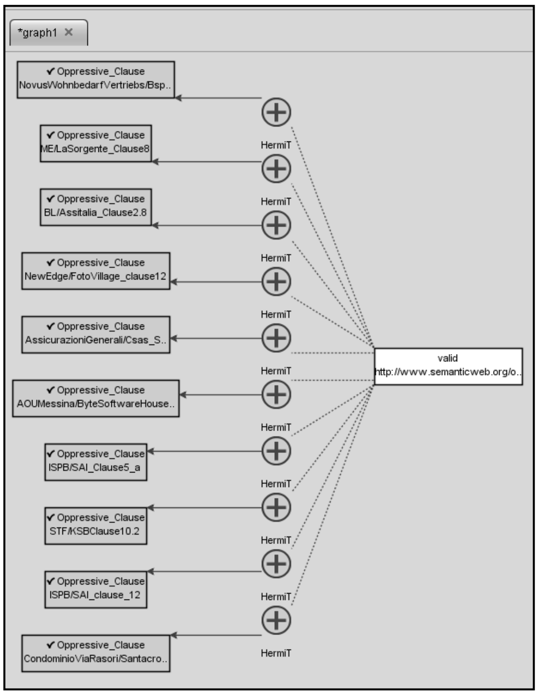


Fig. 4.15 – The query results.

This structure allows to check not only the formal adjudication (if the *accepts* property is already written in the T-Box of the Legal Ontology representing the case at hand), but – thanks to the first rule presented in this section – it also allows to check the *substantial adjudication*, verifying whether the sentence applies the statuses and laws as suggested either parties, or whether it doesn’t. It is however to be noted that this latter function doesn’t involve a complex reasoning but rather represents a kind of advanced “lookup” function.

4.4. The application of Carneades by the present research

4.4.1. Introduction: Querying the

ontology

In the present section, an example of judgement representation is given. The task of the user in this example is to check which cases (i.e. in the presence of which factor, or in which tribunal) an oppressive clause is judges as inefficacious. The first step consists in querying the Carneades System to retrieve a list of contract clauses which have been considered oppressive (either by the parties as an undisputed fact, or following a judicial interpretation). Figure 4.15 shows the resulting list.

In the present example the second result (`ME/LaSorgente_Clause8`) will be further analysed. Please notice that this instance does not represent a legal case, but rather a contract clause. Information contained in the ontology allows retrieving in any moment the details concerning the case (court, date, parties, the decision's text, and so on) but such information will not be shown on the argumentation tree: in it, only simple statements concerning the contract clause, other material circumstances related to it (i.e. the contract it is contained in), legal statuses, and interpretation instances – the statements which are relevant to the dynamics of the judicial argumentation – are found, those who pertain to the identification of the case being retrievable but not shown here.

4.4.2. Reasoning on law applicability

The next step is to ask the system Whether `ME/LaSorgente_Clause8` can be considered *inefficacious* in the light of applicable laws and judicial interpretations made by the judge in the precedent (Figure 4.16 shows the results).

The system found two applicable laws to argue the inefficacy of the clause (left part of Figure

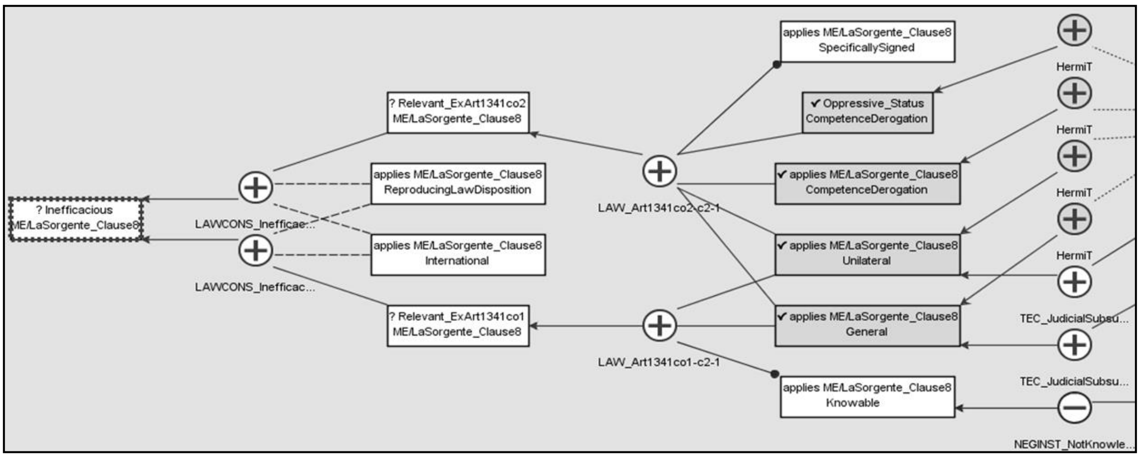


Fig. 4.16 – Argumentation PRO the contract clause being inefficacious.

4.16): if the conditions of one of these two laws are met, and no exception exists (in this case, possible exceptions - broken lines - are the contract being an international contract, and the contract reproducing law dispositions), the clause is inefficacious.

The requirements for a clause to be relevant under one of these two laws are presented in the central part of Figure 4.16:

In order for the clause to be **relevant under Article 1341co1** it must be *general*, *unilateral* and *not knowable* by the other party by using ordinary diligence (following is the rule in LKIF-Rule language).

```
<rule id="LAW_Art1341co1">

    <head>

        <s pred="Relevant_ExArt1341co1"><v>C1</v> falls under
        the discipline of Article 1341 comma 1 of Civil Code
        </s>

        <s pred="&oss;considered_by"><v>C1</v> falls under the
        discipline of <i value="&oss;Art1341co1cc">Article
        1341 comma 1 of Civil Code </i></s>

    </head>

    <body>

        <s pred="&oss;applies"><v>C1</v> applies <i
        value="&oss;General"> general status</i> </s>

        <s pred="&oss;applies"><v>C1</v> applies <i
        value="&oss;Unilateral"> unilateral status</i> </s>

        <not><s pred="&oss;applies"><v>C1</v> not applies <i
        value="&oss;Knowable"> knowable before contract
        sign</i> </s></not>

    </body>

</rule>
```

In order for the clause to be **relevant under Article 1341co2** it must be *general*, *unilateral*, *oppressive* and *not specifically signed* (following is the rule in LKIF-Rule language).

```
<rule id="LAW_Art1341co2">

    <head>

        <s pred="Relevant_ExArt1341co2"><v>C1</v> falls under
        the discipline of Article 1341 comma 2 of Civil Code
        </s>

        <s pred="&oss;considered_by"><v>C1</v> falls under the
        discipline of <i value="&oss;Art1341co2cc">Article
        1341 comma 2 of Civil Code </i></s>

    </head>

    <body>

        <s pred="&oss;applies"><v>C1</v> applies <v>S1</v>
        </s>

        <s pred="&oss;Oppressive_Status"><v>S1</v> is an
        oppressive status</s>
```

```
<s pred="&oss;applies"><v>C1</v> applies <i value="&oss;General"> general status</i> </s>

<s pred="&oss;applies"><v>C1</v> applies <i value="&oss;Unilateral"> unilateral status</i> </s>

<not><s pred="&oss;applies"><v>C1</v> not applies <i value="&oss;SpecificallySigned"> specifically signed</i> </s></not>

</body>

</rule>
```

4.4.3. *Reasoning on judicial interpretations*

The clause was found to be oppressive (since a clause concerning competence derogation is part of the list of article 1341co2), general and unilateral (dark boxes with a tick are accepted statements, dark circles with "+" are valid *pro* arguments). The arguments used to accept these statements are *arguments from ontology*, which means that the relative information has been manually inserted in (or inferred by) the database and it is not possible to further explain those positions. The search, however, was not deep enough to determine whether this clause is specifically signed or not, nor whether it was knowable or not (white boxes represent undecided – stated or questioned – statements).

4.4.3.1. Introducing *pro* arguments

The next step is to ask Carneades to produce argumentation towards the acceptance/rejection of the yet undecided statements. Carneades first searches for arguments *pro* the clause being specifically signed: Figure 4.17a shows that the requirement for a judicial interpretation towards the specific signing of the clause is met: the clause is "correctly recalled" and therefore can be considered as specifically signed following some precedent (unless the "recall exception" applies). Following is the rule:

```
<rule id="TEC_SpecificSigningByRecall">

<head>

<s pred="&oss;applies"><v>C1</v> applies <i value="&oss;SpecificallySigned"> specifically signed status </i></s>

</head>

<body>

<s pred="CorrectlyRecalled"><v>C1</v> is correctly recalled</s>

<not exception="true"><s pred="RecallException"> <v>C1</v> is subject to the exception</s></not>
```

```
    </body>
</rule>
```

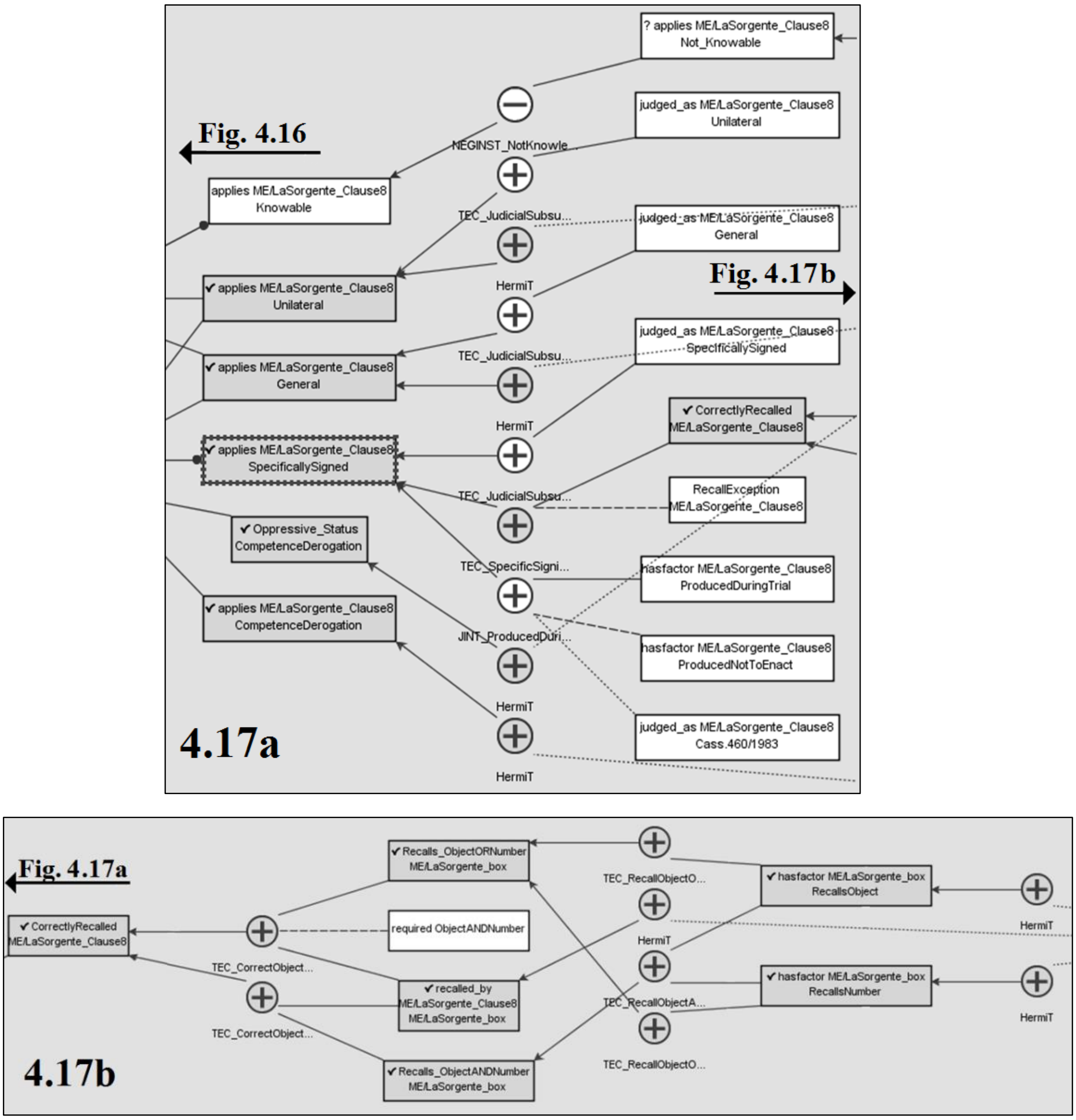


Fig. 4.17 (a-b) – Argumentation PRO the clause being specifically signed.

Figure 4.17b explains why the "correctly recalled" premise has been marked as "accepted" by Carneades: the ME/LaSorgente contract contained a distinct box (usually placed at the end) which recalls object and number of the oppressive clause, and the box has been signed by the other party. It is unknown whether the indication of both object and number in the distinct box is required for the recall to be considered correct in the present case (it depends on the precedents that the judge decides to follow: both may be required, or just one of the two may suffice) but it doesn't matter, as long as the case fulfils the most restrictive requirement of indicating both the object and the number. Following is the rule:

```
<rule id="TEC_CorrectObjectANDNumber">
```

```
        <head>

              <s pred="CorrectlyRecalled"><v>C1</v> is correctly
        recalled</s>

        </head>

        <body>

              <s pred="&oss;recalled_by"><v>C1</v> recalled by
        <v>B1</v> </s>

              <s pred="Recalls_ObjectANDNumber"><v>B1</v> recalls
              object and number</s>

        </body>

  </rule>
```

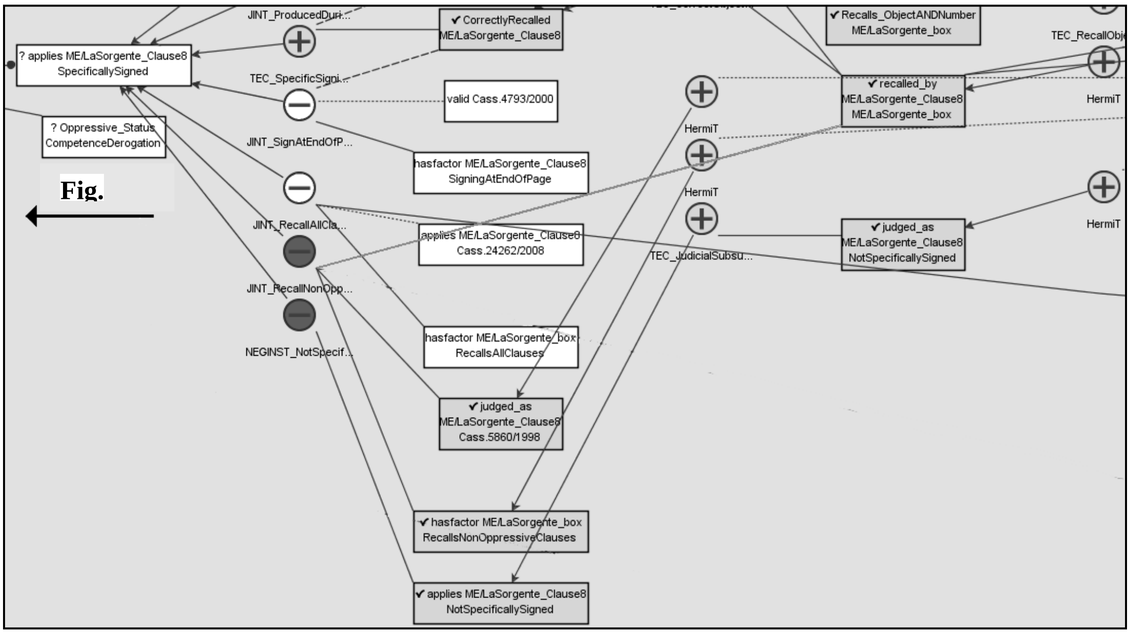


Fig. 4.18 – Argumentation CON the clause being specifically signed.

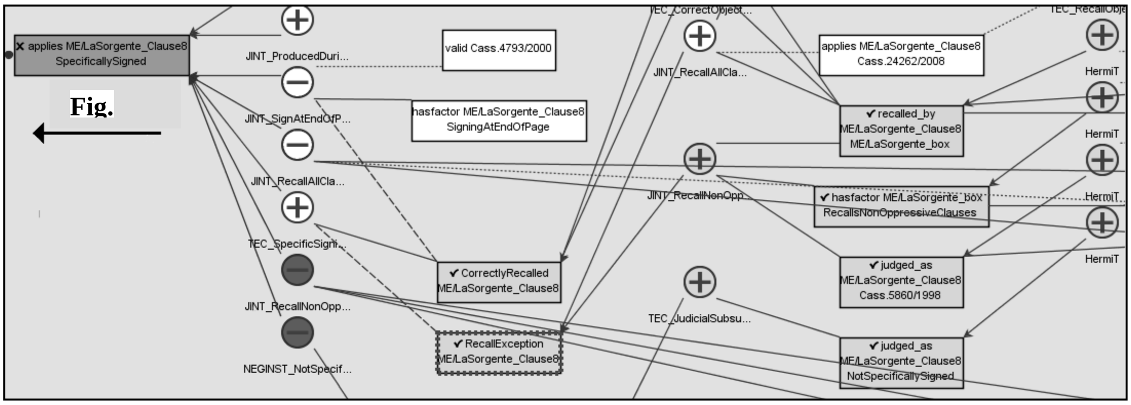


Fig. 4.19 – Argumentation on the acceptation of the exception.

4.4.3.2. Introducing *con* arguments

A position to support the specific signing of the clause was found, the argument being called *Specific Signing Through Recall*, but it is prone to an exception which has not yet been explicitly rejected. Before checking it, it is possible to check if there is some plausible argumentation path leading to the opposite conclusion (*con* the clause being specifically signed). Carneades found two paths leading to the opposite conclusion (Figure 4.18, darker circles with "-" are valid *con* arguments), and therefore the statement "applies `ME/LaSorgente_Clause8 SpecificallySigned`" has turned white – and undecided – again. The bottom argument, called `NEGINST_NotSpecificallySigned`, is a simple instantiation of a negation, turning a positive status *notA* into a negative status *A* (this fiction is necessary to translate between Description logics-, open-world-based OWL ontologies and defeasible rules while keeping full OWL expressivity and should be rendered unnecessary by the implementation of OWL2's Negative Object Property Assertion). Going further backwards, it turns out that the statement "`ME/LaSorgente_Clause8` judged_as `NotSpecificallySigned`" is an *argument from ontology*: it represents a judicial interpretation made by the judge in the case which had that very clause (`ME/LaSorgente_Clause8`) as its object. Hence, the judge actually interpreted that clause as not having been specifically signed – But why? That question is answered by the second argument (`JINT_RecallNonOppressiveClause`): this argument signifies that the box which contains the specific signing contains also clauses which are not oppressive. This, under some interpretation, may render invalid the signing made on the box: in particular, the relevant precedent (Cass.1860/1998, which means Cassation Court decision n. 1860 of 1998) is not only suggested, but thoroughly accepted: this means that the judge, while interpreting the case where `ME/LaSorgente_Clause8` was involved, explicitly cited that precedent in his decision.

4.4.3.3. Reasoning on exceptions

The evidences presented above suggest that the solution taken by the judge was to consider the clause as not specifically signed, but in order to verify the consistency of this one last step is needed, namely, checking if the exception for the *pro* argument towards the specific signing of the clause applies (Figure 4.19, darker boxes with "x" are rejected statements). Following is the rule responsible (with multiple heads) responsible for both the support for the *con* argument and the undercutting of the *pro* argument.

```
<rule id="JINT_RecallNonOppressiveClauses">

    <head>

        <not><s pred="&oss;applies"><v>C1</v> doesn't apply <i
        value="&oss;SpecificallySigned"> specifically signed
        status </i></s></not>

        <s pred="RecallException"> <v>C1</v> is subject to the
        exception</s>

    </head>

    <body>
```

```
<s pred="&oss;recalled_by"><v>C1</v> recalled by
<v>B1</v> </s>

<s pred="&oss;hasfactor"><v>B1</v> has factor <i
value="&oss;RecallsNonOppressiveClauses"> recalls also
non oppressive clauses</i></s>

<s pred="&oss;judged_as"><v>C1</v> applies<i
value="&oss;Cass.5860/1998">precedent
Cass.5860/1998</i></s>

  </body>

</rule>
```

Figure 4.20 shows the complete argumentation graph of Figure 4.16. The oppressive clause has been judged as inefficacious, and the graph shows why (white boxes) it could be that a similar case gets a different outcome.

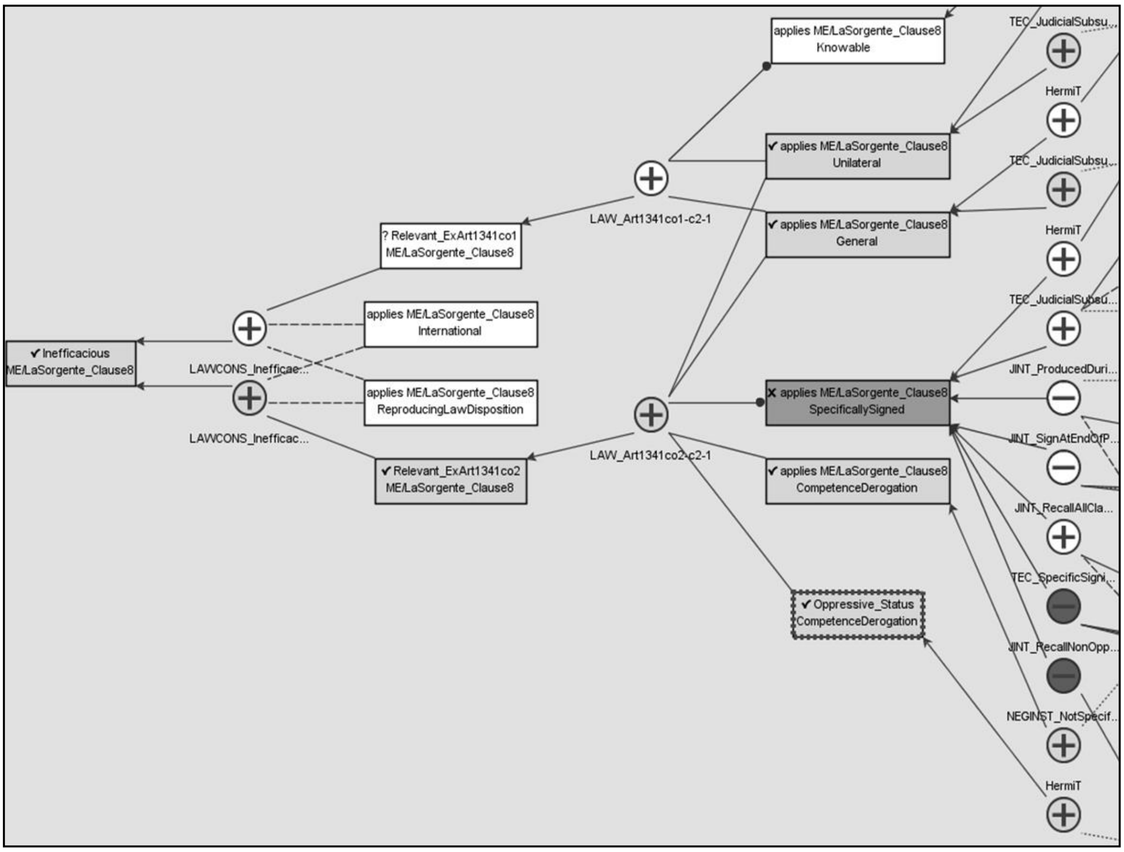


Fig. 4.20 – The complete argumentation graph

4.4.4. Suggesting New Argumentation Paths

Carneades can also suggest new interpretations for known facts, in the light of existing norms and relevant precedents. The system can analyse the relevance of the same clause (ME/LaSorgente_Clause8) under article 1342co2. Under that perspective, the clause still lacks an acceptable argument *pro* or *con* its knowability. Trying to find some arguments *pro* the knowability it turns out that Carneades has noticed the signed box which recalls clause 8. Using a different legal reasoning, he takes into account the rule `JINT_KnownDocumentRecall`, introduced by TribPiacenza2.1 (a decision which is inside the knowledge base of the Domain Ontology):

```
<rule id="JINT_KnownDocumentRecall">

     <head>

          <s pred="&oss;applies"><v>C1</v> is not <i
          value="&oss;Specific">a general clause</i></s>

     </head>

     <body>

          <or>

               <s pred="&oss;hasfactor"><v>C1</v> has factor<i
               value="&oss;RecallsSeparateDocument">recalls a
               separate document</i></s>

               <and><s pred="&oss;recalls"><v>C1</v> recalls
               <v>D1</v></s><s
               pred="&oss;SeparateDocument"><v>D1</v> is a
               separate doc</s></and>

          </or>

          <s pred="&oss;hasfactor"><v>D1</v> has factor <i
          value="&oss;KnownToParties"> is known to the
          parties</i> </s>

          <not>

               <not assumable="true"><s
               pred="&oss;judged_as"><v>C1</v> is judged as <i
               value="&oss;Cass.3929/1999"> precedent Cass.
               3929/1999</i> </s></not>

          </not>

     </body>

</rule>
```

Following this interpretation, if a clause is recalled by a document which is known to the parties, the clause has to be considered knowable. Carneades also found out that the distinct box is a contractual agreement, and therefore presumes (dotted green line) that the interpretation TribPiacenza2.1 can be applied to the case. This specific interpretation of the case ME/LaSorgente was not contained in the decision text (which does not talk about the profile of knowability, and this is testified in the ontology by the lack of judged_as properties linking the clause to one of its I/O statuses) and neither it cites the precedent of TribPiacenza2.1, but nevertheless Carneades suggested this way of argumenting *pro* the target statement (Figure 4.21).

This could give hints on the practicability of such a strategy in a similar case, or arise comments on the difference between the legal concepts of "knowability" and "specific signing" in the terms of the relevance of the number and kind of clauses which are recalled in a separate document.

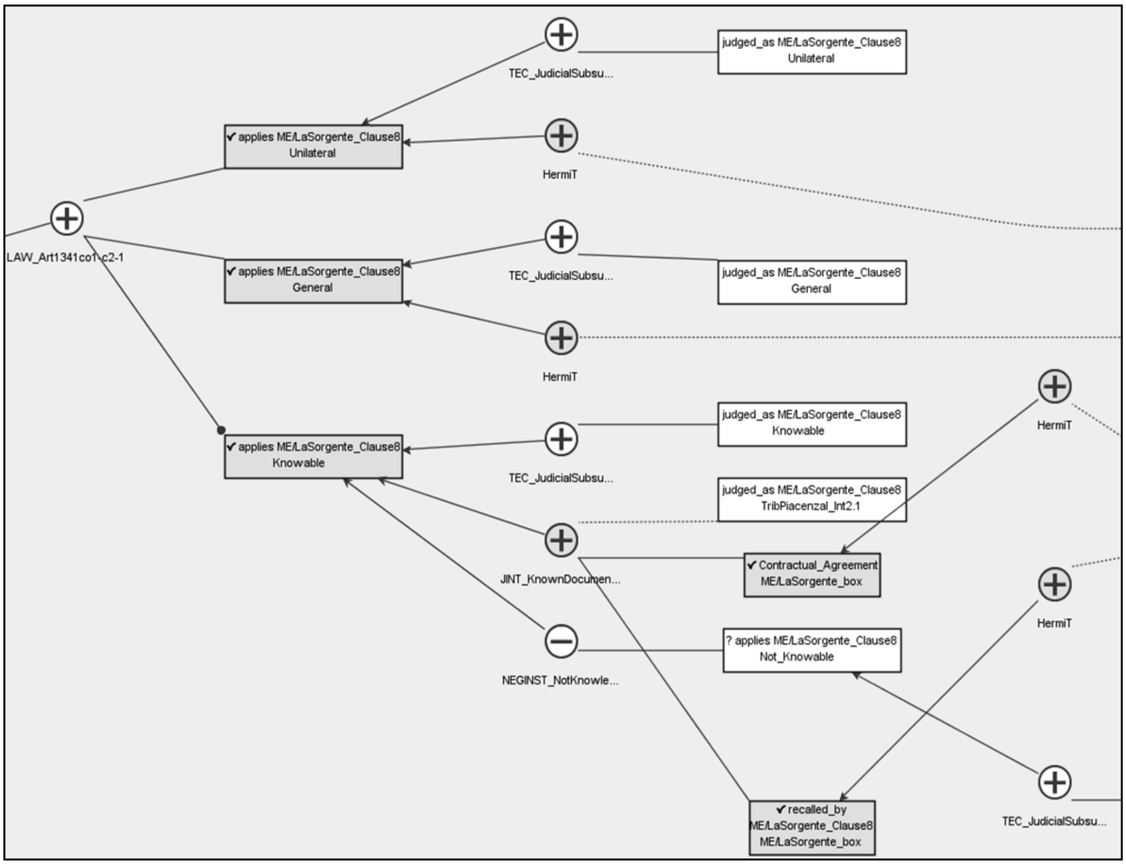


Fig. 4.21 – Suggested arguentation PRO the clause being knowable.

4.4.5. Other Applications

The example only showed how to model one single case, albeit giving the idea of which different directions can be taken from there. In this environment, it is in fact possible to conduct many and more complex activities even with the small number of cases already contained in the OWL knowledge base: it is possible to query precedents (such as `TribPiacenza2.1`) in order to understand the characteristics of the case, and to compare them to other precedents or to a new case; it is possible to investigate the relevancy of a clause having certain characteristics under a specific norm or judicial interpretation, and those characteristics can be either manually inserted as statements in the argumentation graph, or automatically extracted from precedents or knowledge bases (even outside the Domain Ontology). The defeasible logic behind the reasoner and the solid proof standards system allow a complete analysis of possible exceptions to rules and interpretations, through a *positions searching* activity: the system takes - in turns - the part of the attacker and the defendant and produces arguments *pro* and *con* the given statement. These advanced and more automated functions are however hindered by two limits of the 1.0.2

version: the lack of automatic management of weights and proof standards, since arguments did not automatically inherit weights from rules (but this has been overcome in version 2.x, see 4.2.6.1.) and the reasoning times which rise exponentially when incrementing the rule base and / or the ontology KB beyond a relatively small threshold.

4.4.6. Version 2.x

The same experiment was tried with the new version of Carneades, which is still under construction (for a presentation of the features introduced in version 2.x, see above 4.2.6.). The main differences in the approach were the following:

- Although previewed, at the actual stage Carneades 2.x does not include an import engine for ontologies. The knowledge had therefore to be inserted in the rules file, to give the basis for reasoning on the rules to be checked.
- The new version dismissed the XML language in favor of s-expressions.

One of the features in the new rule language which is most interesting in the present research is the built-in hierarchisation of rules, which are divided in sections and schemes. Every section is an individual (identified by a name) but also a composition of schemes, each of which is in turn individually identified. The section element allows also an internal hierarchy: a section may well contain other sections, the top- and bottom layers being ‘theory’ and ‘scheme’, respectively. This feature allows to reach a certain degree of similarity between the interpretation of the legal document and the form of the syntax representing it: it is in fact possible to represent a legal rule using more than one Carneades scheme (because of the complexity of the logics behind the law clause) while at the same time maintaining the correspondence of a single law clause with a single Carneades section. Unfortunately, its level of granularity does not ensure the filling of the gap between representation and reasoning in all situations, as explained in issue 3.3.4.5.

Here is an example of a rule written in the new Carneades rule language:

```
[(make-section
          :header (make-metadata :title "Art1341" :creator "Marcello")
    :schemes
    [(make-scheme
     :header (make-metadata :title "Art1341co1" :creator "Marcello")
     :conclusion (Relevant_ExArt1341co1 ?x)
     :premises [(pm '(applies ?x General))
                          (pm '(applies ?x Unilateral))
                          (pm '(applies ?x Not_Knowable))])
```

```
(make-scheme
     :header (make-metadata :title "Art1341co2" :creator "Marcello")
     :conclusion (Relevant_ExArt1341co2 ?x)
     :premises [(pm '(applies ?x ?y))
                          (pm '(Oppressive_Status ?y))
                           (pm '(applies ?x General))
                           (pm '(applies ?x Unilateral))
                           (pm '(not(applies ?x SpecificallySigned)))])])]
```

The main problem with the new version of the rule engine resides in its schemes being some sort of Horn clauses. As described in 1.2.2.3, a Horn clause is a clause with at most one positive literal. Horn clause logics allows inference to be made substantially more focused than in the case of general resolution: as long as a resolution is restricted to Horn clauses, in fact, some interesting properties appear. This could enhance computability, but unfortunately the limitation of Horn rules to one conclusion limits the balance between interpretation of legal text and expressivity of rules, which can no more represent legal rule with a 1:1 ratio, but must necessarily be modeled as an abstract net of relation where each node can evoke at most one higher level node. This issue constitutes part of the wide problem of mixing a *faithful* representation of a text's semantics with a syntactical model for reasoning on the concepts (see Chapter 5).

4.5. Issues

Importing ontologies, adding factors and writing rules in the LKIF-Rule and Carneades languages highlighted some critical aspects in the modelling process. In particular, the issue was about the correct design and management of information between the ontologies and the rules: some of the axioms already modeled in the ontologies, in fact, could better meet their potentialities if modeled as an LKIF rules instead. The issue, anyway, should be solved with general criteria, since its implications are many and important, starting with the different logic used (description logic for OWL vs. defeasible logic for LKIF). Furthermore, the knowledge imported from the ontology includes the inferred knowledge, but it is not possible to re-use the ontology axioms to obtain new inferences on the basis of the knowledge created by Carneades' reasoner. This suggests the distinction between static information (thesauri, taxonomies, administrative and procedural data) to be included in the ontology, and legally relevant information (legal statuses, subsumptions, inclusion of a material circumstance into the scope of a norm) to be modeled as rules for the purposes of the argument evaluation. It is also possible to enhance Carneades with a system capable of translating OWL axioms into defeasible rules, and Carneades 2.x should include this feature. However, neither this solution is optimal, since it alters the logics underlying SWRL (OWL) rules, therefore creating a significant risk of semantic (or syntactic) shift. This will be better examined in Chapter 5.

4.5.1. Description logics vs. the Carneades framework

While DL is very powerful and useful, monotonic logics are not sufficient for modelling legal rules, such as the rules of consumer law, in an expressive and verifiable way, while at the same time maintaining the structure of the legislation and regulation they are ought to represent: legislation is typically organized as general rules subject to exceptions, and arguments made by applying legal rules are defeasible. Moreover, the application of laws depends on time, and various legal rules may conflict with each other: these conflicts are resolved using legal principles about priority relationships between rules.

Carneades includes proof standards, which according to [Governatori 2011] can be represented through a kind of priority relation in defeasible logics. The new Carneades 2.x also includes, in its rule system, the rule property `<strict>` which allows to specify (through the values true/false) if a `<scheme>` is either strict of defeasible. However, this property does not ensure full expressivity of the defeasible logics constituents, since it represents defeaters or metarules through weights, which should – in theory – represent a more concrete and meaningful way than direct superiority relation (see Chapter 5).

4.5.2. Identifying a border between semantic representation and syntactic modelling.

One of the main goals of the framework built so far was to fill the gap between the text and the reasoning, allowing for knowledge contained in judicial decisions to be semantically enriched (while maintaining at the same time a strong connection to the text) and that knowledge to be processed under rules which are open-textured and fully modifiable by the user. In order to achieve that, a twofold approach was used, which comprehended also the use of different systems, theories and logics. As long as these systems rely on different logics, it will be necessary to define a precise border between these two systems, identifying the common tools as well as those which are exclusive of one of the two.

The **semantic representation system** is centered on the Legal Ontology: expressions, qualifications, media, agents and (in the future) time-related metadata are defined in their properties, and additional resources (such as axioms, inverse properties, and property chains) allow to automatically complete the knowledge extracted from the decision's text, transforming some isolated statements into a consistent network of relations between material circumstances, media, agents, laws, judicial interpretations and claims.

The **judicial argumentation system** is centered on the set of rules. It builds argumentation graphs by creating statements which process general rules (as major premises) and statements coming from the ontology (as minor premise) to create new statements, which in turn can be used for further argumentation. The rules to be represented here are law dispositions, judicial interpretation patterns, and logic operators used to represent common structures in the domain.

Despite the appearances, distinguishing which kind of modelling is required is not straightforward for all kinds of information. For example, the property chain creating the judged_as link between the material circumstance and the legal rule needs to be represented also in the Legal Ontology (so that the reasoners can automatically sort the decisions depending on the interpretation instances contained in it, and this sorting depends very often on which material circumstance is linked to which legal status) but, at the same time, it is not desirable to create an equivalence of type `judged_as = applies` in the Legal Ontology, as this would create an enormous number of inferences which would void any search done using either the first or the second property. Instead, this rule of equivalence is fundamental in the judicial argumentation system to elevate the interpretation instances to the higher rank of "legal bonds", which in turn allow reasoning on law relevancy and applicable legal consequences without losing information on the origin of the bond (factual bonds being directly shown as imported from the ontology, while interpretation instances are firstly imported as `judged_as` relations and only in a second time converted into "applies" relations).

4.5.3. The ontology import feature

Importing statements from ontologies is a key feature in Carneades, since it allows to create argument graphs using not only a knowledge base marked in XML, but also knowledge inferred from that base using OWL axioms. However, the ontology import system in Carneades 1.0.2 has two main weaknesses: it introduces statements from the ontology without distinguishing between stated and inferred knowledge, and it does not allow feeding back statements into the reasoner to infer new knowledge from the ontology axioms.

An approach to the issue, which will be implemented in Carneades 2.x and was already discussed in Stefan Ballnat's Ph.D thesis (see also [Ballnat and Gordon 2010]) and is available as an expansion of Carneades 1.0.2, consists in translating owl axioms into rules. Instead of running the reasoner on the OWL file and then importing a list of statements into the knowledge base of Carneades, it is in fact possible to translate the A-Box of the ontology into s-expressions (rules), while keeping the T-Box in the form of statements. In this way, only the knowledge explicitly stated in the original OWL file would appear as imported by the ontology, while all inferred knowledge would be represented in the graph as one (or more) arguments involving the ontology axiom as the argument, its components as premises, and the inferred knowledge as the conclusion.

Such a system for translating axioms into rules could represent the first step for a complete and reciprocal integration between the two sides of the logic layer of the Semantic Web cake, but a fundamental limit to this possibility is the conflicts which would arise between the different logics involved (first order predicate logic for OWL vs. defeasible logics for the rules). Rules that are designed to work within an ontology could bring to undesired results if forced to operate into a defeasible environment. This issue is related to the precedent, regarding the need to distinguish between semantics and syntax: in fact, the power of syntactic expressivity (managing data, in a *web of data* perspective) would gain from this option, while the need for a proper representation of the document's semantics (in a *web of document* perspective) would be properly fulfilled only if all features of description logics, underlying OWL reasoners, are exploited. This will be better examined in Chapter 5.

4.5.4. The issue of IDs for rules

In order to keep a strong link between the law text and the legal rule language, it is necessary that the rule language allows the assignment of an ID to each rule and its segments, with a sufficient level of granularity. IDs, in fact, can be used to link the statement to the fragment of text which expresses it. Both the LKIF-rule language and the Carneades rule language do allow assignment of IDs to rules.

On this behalf, Carneades 2.x includes two different metadata concerning rule properties: `<source>`, to indicate the source text, and `<header>`, to identify the author of the rule model. These blocks can process data expressed in the Dublin Core standard, allowing to assign an URI and an ID to its values, thus ensuring a full semantic expressivity. It is, of course, possible to assign the same source to different sections or schemes.

Unfortunately, it is impossible to assign different IDs to the body and head of a `<scheme>`. This is due to a precise design choice: Clojure, in fact, allows IDs for parts of code, but Carneades does not have the ambition to reach such a level of granularity. This is albeit necessary in order to maintain a complete adherence of the syntactic structure of rules to the representation of the interpretation of the legal document where those rules (or parts of rules) found their existence: in fact, it is not uncommon that the body of an abstract rule of law is located in an article, while its head is located in the following article, or even in a different law. Without a function to assign IDs to the constituting elements of a rule, it is not possible to ensure a complete and unambiguous system of linking of abstract legal rules to the fragments of law from which they originate.

4.5.5. Tuning of factors, statuses, material circumstances and interpretration instances

In the ontology set, the core mechanism for representing the knowledge is the `qualifies` property, and its subdivision into the `considers` and `applies` sub-properties. This system allows to create bonds between material and abstract concepts, without creating a relation of identity between these elements or confusing the ontology about what is a material element, which are the abstract concepts bound to it, and which kind of link is binding them. In the core ontology, the abstract classes are defined with an open approach, i.e. a legal status can appear on both sides of the triple: it can be the resulting status which is applied to a circumstance by a judicial interpretation, or a requirement for a legal consequence to be applied. Nevertheless, under a taxonomic point of view, the borders between the concepts of legal rule, material circumstance and legal consequence are sufficiently defined, and there is no major risk of misplacement of an element of the judicial decision when classifying it under those categories.

However, delving deeper into the judicial interpretation instances means highlighting not only the legal statuses already identified by the law, but finding new “characteristics” which the court considered as relevant in the process of subsuming a material circumstance with those characteristics under a legal status. It is in that phase that factors come into play. A factor is an abstract concept that defines some material circumstance through some neutral, non-formalized characteristic. Anything can be a factor, as far as it is taken under consideration as a relevant definition for the material circumstance upon which it is being argued.

Under a logical point of view, there is no practical reason for distinguishing between factors and legal statuses, as they play the same role in the axiom and argumentation systems, the only difference being theat they instantiate different argument schemes. However, under a juridical point of view, there is a strong semantic distinction between attributes which are considered as important by the law, and attributes which are taken into consideration by a private agreement or a court. Keeping these two classes will strongly improve queries and searches, which are pointed either at high, official legal concepts or lower, more experience-driven factors. Unfortunately, it is not possible to create a strong distinction between these two classes: in fact, neither *legal statuses* are official, being only a representation of a normative expression contained in the law's text. This means that a characteristic which is created as a *legal status* in some domain, can count as a *factor* in some other domain, or vice versa. The open structure of these concepts does not prevent the system to work even in the lack of a clear distinction of these concepts. However, until a strong policy is found and applied in the classification of the attributes into the *factor* and *legal status* classes, it won't be possible to rely on that semantic distinction between higher and lower level attributes.

A similar issue arises in the distinction between material circumstances and other elements, such as interpretation instances and legal rules. In fact, it can happen that some judicial interpretation takes into consideration some law text or some precedent. In the cases where those elements are taken into consideration in order to apply some status (or factor) to them, they are considered as circumstances: a single event which is being qualified by ascribing it to one or more mental categories. This actually turns them into material circumstances for the purpose of that interpretation instances, because in that case a law text or a precedent is no different from a person's behavior, or an event: they are some external things which are about to receive some qualification, which in turn could subsume them under some secondary axiom which gives them more attributes, and so on.

Also in this case, it will be probably necessary to formalize these cases in order to prevent an uncontrolled spreading of these categorization: for example, if all legal rules and judicial interpretation would acquire automatically the status of material circumstances, this would void the semantics brought by the material circumstance class, and searches and queries involving this concept would bring no significant or very noisy results.

4.5.6. Enhancing scalability: From statuses through factors to elements, from general concepts to details

The layered structure of the Legal Ontology (from legal statuses to factors) allows a controlled and useful redundancy of knowledge on interpretation instances contained in the decision's text, which enhances reasoning capabilities by optimizing the arguments search following a multi-layered structure.

In fact, the search for arguments can be conducted on a step-by-step basis, starting from the top level (looking for applicable laws and major precedents, which take into consideration only legal statuses) and then specifying the statements introduced by the reasoner with a direct search trying to justify the interpretation of some material instance as a specific status. Please notice that it makes no difference whether the statement being examined is accepted, rejected or questioned in the knowledge base: the argumentation system will look into lower-level rules which have the required statement as head (conclusion) and build a graph further dissecting the issue, proposing possible paths towards the acceptability of the assertion concerning the legal status.

The main advantage of this layered structure is the possibility to represent the decision at an exact *depth level* (more precisely at the exact depth level reached by the court) without affecting the chance to build, starting from it, argumentation hypotheses at a shallower or deeper level. For example, if the decision simply stated that "a clause signed at the end of page is not specifically signed", without specifying if some recall was made at the end of the page, the information can be represented at a simple *legal statuses* level. However, if another decision (or a law, or technical considerations) further define the concept of *end of page*, i.e. taking into consideration the emphasis made on the signing at the end of each sheet, this kind of information will be coded at a deeper level but linked to the same "end of page" concept. Each of these two precedents will then appear when arguing about the other one, either at a shallower or deeper level of reasoning (in the first case, the system will simply propose to accept the precedent excluding the specific signing, while in the second one the system will propose to check the existence of an emphatic recall at the end of page and then to accept the corresponding precedent).

Even a bottom level could be built, representing material circumstances in terms of linguistic phenomena. For example, the formula of a "distinct box recalling oppressive clauses" could be modeled, in order to allow NLP tools to automatically spot this kind of text and suggesting the existence of such a factor (or legal status) on a contract clause.

4.5.7. The tuning of properties

Parallel to the issue of the factor/status distinction, stands the issue related to the modelling of properties which link a material circumstance to its attributes. In fact, the property chain `judged_as` constitutes a model of many possible "inferred links" which bring forward compact and important semantics for the argumentation to use. In order to represent the kind of link created by the consider/applies combination it could in fact be sided by a `has_factor` property, a `agreed_as` property, and so on. Unfortunately, the current semantics for property chains does not allow to restrict the type of instance which "stands in the middle" of the chains (i.e. `judged_as` only if the "middle" instance is of type `Judicial_Interpretation`, `agreed_as` only if it is a `Contractual_Agreement`). This leaves two possible solutions:

- Defining a *basic* property as a result of the chain, then further specifying it with subclasses automatically assigned by the reasoner through appropriate axioms;

- Distinguishing `considers` and `applies` properties through the creation of specific subclasses for some subclasses of qualifiers, which can then be automatically inferred through axioms.

Unfortunately, both solutions have the problem of greatly increasing the quantity of data to be managed by the model, thus compromising its reasoning performances.

4.5.8. Reification of negative properties in the ontology

When creating the Core and Domain ontologies, in order to maintain the semantics of negative properties (ex. A contract clause is not specifically signed), this negative assertion has to be reified by creating a specific status (i.e. `not_specificallysigned`). This choice was forced by the open world assuption used by the DL standing behind computational ontologies. In fact, when working with classes (e.g. representing the *specifically signed* status with a class `Contract>SpecificallySignedContract`) would have allowed the representation of a negative attribute by introducing axioms with the `not(SpecificallySignedContract)` variable in it, but in no way would it have been possible to represent this semantics when dealing with a triple of type

```
ObjectProperty a b
```

In the present application of Carneades 1.0.2, however, the original semantics of negative attributes is restored, through some “rules enhancing semantics” such as:

```
<rule id="NEGINST_NotSpecificallySigned">
    <head>
        <not><s pred="&oss;applies"><v>C1</v> applies <i
            value="&oss;SpecificallySigned">     specifically
signed status           </i></s></not>
    </head>
    <body>
        <s pred="&oss;applies"><v>C1</v> applies <i
            value="&oss;NotSpecificallySigned">  specifically
signed status       </i></s>
    </body>
</rule>
```

In this way, the argumentation rules (and consequently the graphs they create) can directly refer to the semantics of positive or negative properties, referring to them with the formulas `C1 applies SpecificallySigned` and `not(C1 applies SpecificallySigned)` respectively. This allows full semantics representation in the argument construction and evaluation phase, including the correct management of presumptions and exceptions. It is also possible to use Boolean operators (true/false) for representing this, but since the reasoning involves both rules and ontology, which have a different approach on "negation as failure", the solution could create undesired results. Setting both the positive and negative statements as a positive status, instead, ensures that the same requirements apply for these statements to be matched.

OWL 2.0 introduces a new feature called *Negative object property assertion* which, in the form `NegativeObjectPropertyAssertion(OP a b)` states that the individual *a* is not connected by the object property expression *OP* to the individual *b*. This will allow the semantics to be fully represented also in the Core and Domain ontologies, removing the need for instances reificating negative properties. The present research revealed that when representing knowledge in a logic language with an open world assumption, it is always better to instantiate all entities, including negative ones. This because the aim is to distinguish between cases where no proof of the positive status is present from cases where the proof of the negative status was reached. Moreover, proving the lack of some circumstance in a trial is as relevant as proving its presence (for example, see Carneades’ system of burden of proof, where the proof standard (or status) for the negation of the target statement can be derived if know the proof standard required for the acceptance of the target statement. Since stating that some material circumstance has a certain negative property is semantically different from stating that some material circumstance does not have a certain positive property, and the two statements are treated differently by OWL reasoners when deriving new knowledge, the present research will keep reificating negative statuses, properties and classes at the ontology level.

4.5.9. Resolving conflicts between rules: weights

The modelling of the hierarchical structure of rules is a fundamental step towards a complete representation of the legal argumentation process. Without an instrument to manage conflicts between rules, it would be impossible to apply more than a few rules at a time, and to produce argumentation graphs involving different disciplines. Even outside the field of law application, a hierarchy between rules is fundamental to represent the relative strength of precedents (depending i.e. on the court who issued them, or on the number of citations, or on the date of issue) and therefore to enhance the *find positions* process. Finally, it allows enriching the argumentation scheme with detailed, technical rules or minority reports which – once given a small hierarchical power – can successfully represent minor pieces of reasoning without them affecting the "major" reasoning process (that involving law rules or precedents).

The Carneades System allows the assignment of *weights* to arguments, which is exactly the system needed to give an absolute hierarchy to rules which allows the same value to be assigned to more arguments. Unfortunately, Carneades 1.0.2 does not allow to assign a weight to a rule (thus transferring that value to any argument created through that rule), but only to manually specify the weight of single arguments in the graph. Fortunately, Carneades 2.x overcomes this limitation through an integration of automatic weight assignment and manual customization. The new Carneades Rule Language, in fact, allows to assign weight to a scheme and this weight will be transferred to any argument built using that rule. At the same time, it is possible to manually assign weights to single arguments through the GUI. This function is mostly used in the Policy Modeling tool, where assigning weights to competing arguments is the main way to reason on these abstract policy proposals, and the weight represents the personal persuasion on the efficacy of the arguments. In an environment such as the present one, where weight should represent the authority of the legal act, weight should be assigned in the light of metarules modelling the hierarchy of legal sources. Therefore, it should be possible not only to specify the weight of the single scheme, but also to derive the weight of the argument from the application of specific rules which analyze the source of the scheme and every other relevant aspect (time, object, presence of empowering acts and so on).

While trying to reduce Carneades' argumentation structure to defeasible logics, Governatori introduced the concept of weight mapping: a function that takes as input a weight function and produces a set of superiority relations [Governatori 2011], which could constitute one of the useful additions to the argumentation system. Following is a short presentation of the function:

Definition of weight mapping: Given a Carneades Argument Evaluation Structure S = <*Arg*, *Ass*, *W*, *PS*> (where *Arg* is an acyclic set of arguments, *Ass* is a consistent set of literals, *W* is a weight function assigning a real number in the interval [0,1] to every argument, and *PS* is a function mapping propositions to proof standards) and the threshold α, β, γ (as defined in 3.2.5), let $a = \langle P_a, E_a, c_a \rangle$ and $b = \langle P_b, E_b, c_b \rangle$ be two arguments in *Arg* such that $c_a = {\sim}c_b$. A weight mapping is a function *mwgt* that takes as input a weight function and produces a set of superiority relation as follows:

marg(a) >pe marg(b) iff W(a) >W(b);

marg(a) >ce marg(b) iff

1. W(a) > a and

2. W(a)-W(b) > b;

marg(a) >bd marg(b) iff:

1. W(a) > a and

2. W(a)□W(b) > b and

3. W(b) < g;

>dv = 0.

This representation is an interesting translation of the concept of argument weighting in a more formal language, but does not constitute an alternative to the argumentation theory. The idea of replacing argumentation theory with defeasible logics is indeed intriguing, but does not constitute a step towards the filling of the gap between legal documents and computer-based management of their semantics: in legal reasoning, in fact, several different aspects of argumentation theory are involved. The whole process of introducing arguments *pro* or *con* a given statement, for example, is to be viewed not only in its relational aspects but also in its procedural aspects (argumentation as a dialogue) which identify several steps and different patterns or tasks in the argumentation process and in its single actors. The concept of "burden of proof " is strictly related to these dialogical aspects, and being weight mapping dependant from burden of proof, it is impossible to describe it in defeasible logics without a support from argumentation theory, and in particular from procedural aspects of it. For a complete discussion on the subject see 4.6.

4.5.10. The need for a temporal model

Temporal coordinates play a fundamental role in legal systems, and the temporal attribute for laws (and precedents) is qualitatively different from other attributes, or statuses, of normative entities: it has to do with the efficacy of laws themselves, and may determine the applicability of a law in place of one other, or of a different version of the same law. Moreover, there are three temporal coordinates to be taken into account for each rule (time of force, time of efficacy, time of application) and a proper diachronic modelling of legislation should feature all of them, in order to provide a complete analysis on the effects of combinations of norms [Palmirani et al. 2011a]. In fact, when examining a factual circumstance under a legal perspective, the time factor matters twice: it matters for the time of the fact, but also for the time when the fact is being analyzed under the law – this is particularly true in criminal law (where the *favor rei* principle exists). Moreover, legal phenomena such as the *vacatio legis* and the *suspension* periods force to distinguish between the time of force and the time of efficacy of legal expressions, and to detach them from the time of existence of its material support.

If it is true that every system which manages legal rules should feature a complete model for temporal coordinates, this is even truer for a system such as the judicial argumentation system of the present research: in fact, building a collection of different decisions taken in different points in time necessarily involves a variation in the applicable rules. Without a proper modelling of the three temporal axes, it is not possible to compare different decisions, preventing any possible reasoning on two remote precedents even when they share other characteristics such as the object of the claim. The solution of creating different rules for different versions of the law, apart from requiring a huge amount of work and leaving the problem of the plurality of temporal axes unresolved, represents a falsification of the original law expressions which compromises the 1:1 ratio between representation of the legal document and syntactic modelling of its normative content. The norm should always remain unique, while temporal coordinates should ensure the crop of the applicable rules on a case-by-case basis.

At the current stage, the rule system adopted in Carneades 2.x lacks modelling of temporal coordinates for laws. The adoption of the Dublin Core standards could allow representation of temporal information through the “coverage” element, which allows delimiting the spatial and temporal coverage of a resource. However, this does not allow performing advanced reasoning at the level of declarative semantics, but only to conduct queries on the metadata or to write metarules which take these properties into consideration, using the *procedural attachment*, which doesn’t consider the semantics of the values but treats them as literals. It is possible to further formalize the *coverage* element by building a specification of the Dublin Core language. Any further step towards the formalization of a temporal model for argumentation systems should instead rely on the last emerging standards such as LegalRuleML [Palmirani et. al. 2011] representing the path towards a standardization of the logics layer in the Semantic Web cake (see 2.4.6.3. and 5.6.).

4.5.11. Deontic operators.

Carneades also lacks proper deontic logics representation. There is no explicit modelling of deontic operators, even though it is possible represent them syntactically via modals such as:

```
(obligated (transfer ?x))
```

Unfortunately, it is not possible to process this information for deontic reasoning, since deontic reasoning is not implemented in Carneades, but it is possible to model its constituents in a rule-like equation such as:

```
Permission B = (not (obligated (not P)))
```

The importance of deontic logics in legal reasoning, and the issues concerning its representation in a legal argumentation system, will be discussed thoroughly in Chapter 5.

4.6. Relational vs. Procedural aspects of argumentation

4.6.1. Introduction

When trying to resolve some of the issues presented above, it is argued that the various aspects of argumentation can be properly represented through logics: in particular, [Governatori 2011] shows how proof standards proposed in the Carneades framework correspond to some variants of defeasible logics, which could imply that an implementation of defeasible logics is able to compute acceptability of arguments. However this doesn't seem to be the case, in the light of the following considerations.

Logics provide abstract formulas to represent relationships between concepts. With fine tools such as defeasible logics it is possible to successfully represent the complex relations between legal rules, but can they manage the application of these rules, which represent a fundamental step towards the computation of the acceptability of an argument? In theory it could be the case, since the act of *substitution* of abstract symbols in formulas with the values of the situation to be computed should be an automatic process where it doesn't matter which material concept is added as the interpretation of that abstract symbol. For example, if we have a + b = c, we can interpret this simple rule as meaning many different things (for example a=1, b=2, c=3, +=addition, or a=blue, b=yellow, c=green and += mix) and this would not affect the truth function of the equation. On the contrary, in the concrete legal field the single elements bring with them particular conditions (minor rules or metarules), assumptions, exceptions, values, which can significantly alter the outcome of the abstract formula representing the rules.

4.6.2. The Issue

Most of the application scenarios of legal rules are centered around dialogues with two or more parties, in which claims are made and competing arguments are put forward to support or attack these claims (this includes judgements, which are the focus of this paper, but also parliamentary debates and other legal acts). Following Walton (see 4.2.1.), there are several kinds of dialogues, with different purposes and different protocols. This view of arguments as dialogues (or *processes*) contrasts with the mainstream, relational conception of argument in the field of computational models of arguments, typified in [Dung 1995] (see 1.3.2.1.) where argumentation is viewed not as a dialogical process for making justified decisions which resolve disputed claims, but as a method for inferring consequences from an inconsistent set of propositions. To see the difference between these conceptions of arguments, notice that a proposition which has not been attacked is acceptable in this relational model of argument, whereas in most dialogues a proposition which has not been supported by some argument is typically not acceptable, since most protocols place the burden of proof on the party which made the claim.

Following the mathematical model of Doug Walton's philosophy of argumentation and Aristotle's classification, Gordon in [Gordon and Walton 2009] describes argumentation as being divided into three layers: *logic*, *dialectic* and *rhetoric*. While logics deal with the so called *relational* aspects of argumentation, dialectic directly addresses the *procedural* aspects of it. In the light of this first distinction, the claim that Defeasible Logics can manage the acceptability of arguments appears to be an effort to flatten the representation of the first two layers into mere logics, which does not seem to take into consideration the difference of tasks evoked by different argumentation patterns (or argumentation schemes - see above), nor the dialectical (or *procedural*) aspects of argumentation.

In [Gordon 2008] it is argued that legal reasoning is not only deductive, because legal concepts cannot be fully defined by necessary and sufficient conditions. Legal concepts which are defined this way are but hypotheses or theories which cannot be mechanically followed using deduction when one tries to apply these concepts to decide legal issues on concrete cases. As [Hart 1961] puts it, “Legal concepts are open-textured”. The process of determining whether the facts of a case can be *subsumed* under some legal concept is, in fact, the process of argumentation.

> «Contrary to some popular notions, law is not a matter of simply applying rules to facts via modus ponens, for instance, to arrive at a conclusion. Mechanical jurisprudence, as this model has been called, is somewhat of a strawman. It was soundly rejected by rule skeptics like the realists. As Gardner puts it, law is more *rule guided* than *rule-governed*.» [Rissland et al. 2003]

Robert Alexy's discourse theory of legal argumentation explains how judicial discretion can be restricted without resorting to mechanical jurisprudence or conceptualism. In the early works of AI & Law on the subject, argumentation was modeled as deduction in a nonmonotonic logic, i.e. as a defeasible consequence relation. The Pleadings Game - introduced in [Gordon 1994] - still uses nonmonotonic logics and in particular defeasible logics to represent legal reasoning, but these logics are have a procedural layer on top of them which treats the whole argumentation as a process, with a sequence of *moves* by the players which are affected by the precedent ones. Moreover, the task of that game is not that of winning a claim, but that of identifying the main issue of what Toulmin in [Toulmin 1958] defines as *substantial arguments*.

Also the abstract framework for argument-based inference formalized in Prakken's work described in [Prakken 2010] combines Dungean semantics with structured argument. The framework is suitable for modelling reasoning with argument schemes, which are used to reveal implicit premises and to identify specific possibilities of undercutting the argument through what Prakken calls *domain-specific defeasible rules*. The concept of *accrual of arguments*, introduced in [Prakken 1995], distinguishes between arguments whose weight can be added when trying to prove a conclusion, and argument whose weight does not sum up. If such a distinction is not made, the application of mere logic formulas to compute these arguments will lead to inaccurate, even counterintuitive results.

Concerning procedural aspects, Walton's argumentation theory identifies a sequence of stages in a dialogue-like argument, where in each stage some *moves* are allowed to the player (as in The Pleadings Game) and those moves influence the possibilities for further stages. In particular, the concept of stages of the argumentation process is fundamental for the allocation of the burden of proof, which brings the discourse back to the first consideration of this section about the relationship between Dungean Semantics and the dialogical conception of argumentation contained in [Walton 1998]. Proposable exceptions, tacit acceptance, second grade preclusions, irrelevance: logics alone, no matter how powerful, cannot properly evaluate the acceptability of those argument if they cannot identify the stage of the process at which those arguments are introduced and consequently correctly allocate the burden of proof on one of the competing parties, and this in turn is not possible without a dialogical (procedural) conception of argumentation. Defeasible logics can effectively manage complex interaction of rules such as the concept of proof standards. But an argument is much more than just rules, and representing the tasks and patterns presented above by relying only on a set of rules would require a huge effort, and yet produce a complicated and ungovernable result. This is, because these rules would have to simulate dialogical characteristics of argumentation, which are very different from relational ones.

In the Pleadings Game, argumentation was viewed procedurally, as dialogues regulated by protocols, but this was accomplished by building a procedural layer on top of a non-monotonic logic. In LKIF, the relational interpretation of rules is abandoned entirely, in favor of a purely procedural view, and is thus more in line with modern argumentation theory in philosophy

[Walton 2006] and legal theory [Alexy 1989]. Argumentation, as [Gordon 2008] puts it, “cannot be reduced to logic”.

In the Carneades Application, therefore, argumentation schemes are managed in an upper layer than rules. In this perspective, rules are just one of many sources for argument construction along with ontologies (OWL) and cases (Cato, see 1.4.2.2.), whose different logics and formats are translated and mixed into an *argument graph* [Gordon 2010]. The architecture used to instantiate these sources into argument schemes is presented in [Gordon 2011a].

4.6.3. Two Examples

The demonstrations based on famous U.S. courts precedents, such as those presented in 5.4., are aimed at modelling arguments starting from the legal concepts, and show how the reliance on argument schemes and competency question is necessary in order to achieve a reconstruction of the original arguments and to correctly evaluate them. However, those tests do not pay attention to the connection of those concepts with the metadata contained in the source legal document. This is also the approach of Ashley's seminal contributions to the subject: the systems presented in [Ashley, 1991] and [Aleven, 2003] are in fact oriented to the teaching of argumentation in law classes, rather than to the performing of automatic reasoning on the metadata contained in the legal documents.

The approach of the present research is more practical, as described in 3.1., and this approach will be kept also in finding evidence of the need for a modelling of the procedural aspects of argumentation into the emerging rule standards. It appears that the modelling of argument schemes is the only viable choice to properly perform legal reasoning, and for this purpose the concept of argument scheme should include templates which represent procedural aspects of legal processes (such as the acts available to the parties during a court trial). The two examples that follow are in fact taken from the same sample of 27 decisions concerning Consumer Contracts, which constitutes the knowledge base of the research described in the beginning of this paper.

4.6.3.1. First Example

The first example is the decision issued on October 31st, 2006 by the First section of the Tribunal of Salerno, concerning the acceptability of an arbitration clause contained in a public statute (the statute of the Italian Football Federation). The argument put forward by the defender is that the judge is incompetent, since the litigation had to be settled by means of an arbitration following art. 24 of the statute. The argument, however, was presented to the court only at a late stage of the trial. The judge, therefore, specifies that if the claim was formally qualified as a request for “competence regulation” (as the defender himself defined it), it would be inacceptable since those kind of claims can be presented only in the early stages of the trial. The judge, however, decides that the claim concerns the object of the trial, not a competence regulation. Therefore, the claim is acceptable and the judge declares his incompetence in favor of the arbitration court indicated in the statute.

In LKIF it is represented like that:

```
<statements>
```

```
 <s id="s122"> applies TribSalerno_I Judge_not_competent</s>
 <s id="s222">considers TribSalerno_I StatutoFIGC_clause10</s>
 <s id="s322">contained_in StatutoFIGC_clause10 StatutoFIGC</s>
 <s id="s422">contains StatutoFIGC StatutoFIGC_clause24</s>
 <s id="s522">applies StatutoFIGC_clause24 Arbitration</s>
 </statements>
<arguments>
  <argument id="arg1">
   <conclusion statement="s122"/>
   <premises>
     <premise statement="s222"/>
     <premise statement="s322"/>
     <premise statement="s422"/>
     <premise statement="s522"/>
   </premises>
  </argument>
</arguments>
```

The LKIF syntax doesn't permit to be dynamic and the sentences are manually applied to the rules during the argumentation modelling. With the Clojure the expressiveness was enhanced including dynamicity with the rules, boolean operators and also meta-schema of argument to apply:

```
(def a1 (make-argument
  :header (make-metadata
  :description {:en "the judge of Salerno is incompetent, because art. 24 of the FIGC statute introduces a mandatory arbitration for all claims concerning the statute" })
  :scheme "competency regulation"
  :conclusion (not (applies TribSalerno_I Judge_not_competent)
  :premises [((considers TribSalerno_I StatutoFIGC_clause10))(pm '(contained_in StatutoFIGC_clause10 StatutoFIGC))(pm '(contains StatutoFIGC StatutoFIGC_clause24))(pm '(applies StatutoFIGC_clause24 Arbitration))]
```

This is a typical example of how procedural aspects of the legal argumentation can influence the outcome of a claim, and therefore the application of rules (an more generally the logic layer) has to take into account these aspects, in order to achieve a correct evaluation of the acceptability of arguments.

4.6.3.2. Second Example

A second example can give some hint on how arguments, even arguments from legal rules, can be introduced in the judgement for tasks different from that of applying the rule contained in the legal norm. In the decision given by the Tribunal of Rovereto on July 13th, 2006, an article of the Civil Code concerning oppressive clauses, which lists them by subject and considers as oppressive all clauses introducing “a limitation in concluding certain contracts with third parties”, is used as an argument to prove that “there is a general disfavor in the system towards all pacts introducing limitations to competition”. The argument of the oppressive clause is used together with the argument coming from article 81 of the EC Treaty, which explicitly forbids such pacts.

We can see how, in this case, the article of the Civil Code is evoked in the decision's text, and must therefore be marked up and linked to the text of the law. The reasoner should however be aware that this rule doesn't have to be used for its general purpose, which is defining an oppressive clause, but rather for the purpose of supporting the statement that “there is a general disfavor in the system towards pacts introducing limitations to competition”. This can be achieved by defining a framework for argumentation and by introducing argumentation schemes. In the example, the argument involving the article of the Civil Code would not be an argument from legal rules, but rather an argument from authority (or similar), and therefore the article of the Civil Code would not be transformed into an argument by translating the logic form of the rule it expresses, but rather by referring to the authority of the Civil Code and of the institution that issued it (which in this case is the Italian Parliament).

4.7. Conclusions

The Carneades application presented in this chapter was intended to show how an argumentation system can be used to process semantic data in a complex way. The arguments construction and the rules representing code- and case-law could never meet their full potentialities if not supported by a semantically rich knowledge base, such as the Legal Ontology Set presented in Chapter 3. The application is thus a demonstration of how a shared logics and syntax for legal rule representation, combined with a standard core ontology for legal concepts, would constitute the ideal starting point for a totally new conception of case-law classification, browsing and management.

The application represents an advancement of the state-of-the-art in the field because it creates a complete juridical environment performing a real benchmark of Carneades' capabilities: the sample (constituted by 27 precedents) has been completely represented in the ontology set and in the rules, thus heavily stressing the Carneades and OWL reasoners and showing their limits in terms of computability. Moreover, the ontology set used in the present application was not specifically modeled upon Carneades, rather representing an effort towards a standard representation of legal text's contents which ensures isomorphism with the source document and interoperability with different applications in the rules and logics layers.

The present chapter showed how the present Judicial Framework achieves a good representation of judicial argumentation, both in his static aspects (classification of the judgement and of the other legal acts involved) and in his procedural development, thanks to a mix of OWL-DL, defeasible logics and argumentation theory. The next chapter will delve deeper into the issues presented in 4.5., providing examples of how they can be faced and laying down a set of requirements for the ideal rule base and logic reasoner.

Chapter 5

Enhancing Legal Semantics

> "Language is a process of free creation; its laws and principles are fixed, but the manner in which the principles of generation are used is free and infinitely varied. Even the interpretation and use of words involves a process of free creation."
>
> – Noam Chomsky, *Language and mind*.

> "I will argue that in the literal sense the programmed computer understands what the car and the adding machine understand, namely, exactly nothing."
>
> – John R. Searle, *Minds, brains, and programs*.

5.1. Introduction

The judicial framework for semantic enrichment of judicial decisions and for defeasible reasoning on the decision's motivation described in chapters 3 and 4 highlighted several critical aspects in legal knowledge representation, some of which were already known to the literature of the field. The present chapter makes a sum of the critical aspects met so far, and elaborates them in order to lay down a set of requirements for the ideal legal rules language, while the next chapter is dedicated to hybrid reasoners, which exploit both the legal rules and ontology languages. Two projects in the field are analyzed in the light of the requirements.

The present chapter will also analyze different approaches to the representation of judicial knowledge. In it the following topics will be therefore presented:

– **Open vs. closed world assumption**, analyzing feats and flaws of these alternative characteristics of logic languages;

– **Data representation**, trying a mediation between the two conceptions of the web of knowledge bases: the original web of documents, and the new web of data;

– **Advanced legal concept representation**, showing how to model the effects of a complex legal figure (suspension of efficacy) in order to support NLP;

- **Representation of case-law,** taking a classic US case, *Popov vs. Hayashi*, and analyzing four possible representations by the major AI&Law experts, to underline differences in the proposed solutions and to see how they can be combined – or improved.

Successively, a summary of the requirements for a rule markup language as identified during the present research is presented. It is divided into two sets of requirement: ontological requirements, for the correct binding of the abstract concepts to the part of the original text, and syntactical requirements, for the correct representation of the connections between abstract legal concepts.

Finally, the LegalRuleML language is presented. This language is a work-in-progress, would-be standard language for legal rules. It contains most of the features discussed in the present chapter, and therefore it is taken as the starting point for the ideal legal rules language. The presentation also highlights where the standard built so far fails to capture aspects and semantics of judicial documents, as identified by the present research.

5.2. Open world assumption vs. Closed world assumption

The present research delves deep into the state-of-the-art of information technologies for representation of knowledge and reasoning. This implies a lot of logics being involved, and of different types. The galaxy of logics was presented in Chapter 1. The present section is focused on the two kinds of formal logics which are mostly used in the Semantic Web: *description logics*, used for knowledge representation, and *defeasible logics*, used for reasoning and argumentation. These logics have several differences, the most important being that between the *closed world assumption* of defeasible logics and the *open world assumption* of description logics (see 2.2.5.1.).

In the legal field, the concept of truth can be substituted with that of legal validity, thus *negation as failure* would imply believing invalid every predicate that cannot be proven to be valid. This is very useful when performing legal reasoning: when comparing a set of precedents to find which legal outcome seems legitimated by the case-law, it is necessary to restrict the answer within such incomplete information. On the contrary, assuming an open world is extremely useful when representing the legal documents. In this way, the concepts can be classified with the maximum rate of enrichment. With the open world approach, in fact, it is sufficient to create an item “judicial decision” for it to be subsumable under virtually any sub-category of judicial decisions. So, until negative attributes (or an attribute which is disjointed with another) are included, this item will remain available to the information retrieval system. This allows for incomplete databases to be automatically populated with additional statements by the description logic-based reasoners, and this without ambiguity about whether a specific statement derives from explicit information in the source legal documents, or that statement was generated by the ontology reasoner in the lack of opposite indication.

The ideal solution seems therefore to be a combination of the two logics. In this way it is possible to perform an effective argumentation on a knowledge base which is maximally enriched by the ontology reasoners. However, in trying to perform this combination, several issues arise. In particular, as seen in Chapter 4, some statements which are useful when sorting the legal knowledge base become an obstacle when performing argumentation. It would be therefore necessary to “clean” some of the inferences from the ontology reasoner before passing to the argumentation. But how is it possible to distinguish between the useful inferences (those who provide the necessary shallow-level reasoning which allows the modularity of the ontology and the rule set) from the undesired ones? It is also important, as already noticed in Chapter 4, that the translation of enriched knowledge should work in both ways: not only from ontology to argumentation, but also from argumentation to ontology. In this way, the classificatory functions could be completely demanded to the ontology layer, therefore achieving an efficient distinction of functions between the semantic layers, without losing any information which could emerge from the application of the defeasible rules.

The integration of these two logics into a rule language which polishes the inferences of OWL-DL, while at the same time preserving the useful part of the knowledge enrichment they produce, requires hybrid reasoning capabilities. Therefore, after having set out (in the rest of this chapter) the requirements for such an integration, Chapter 6 looks forward to hybrid reasoners for a solution in that sense.

5.3. Web of documents vs. web of data

In the first years of internet, the *web of documents* provided the necessary meta-model for knowledge interconnection through an extension of the classification methods adopted before the revolution brought by ITs. Counter posed to this approach is the arising approach of the *web of data* (presented in 2.1.1.), where the conception of document as distinct information container is set aside and the chunks of information (the pieces of content) are under the spotlight as the pivot of data management.

The web made of documents is the first version of the World Wide Web, and thus, as for all initial phases, it looks like it is deemed to be overcome in time. It is, however, not that simple: the ITs have reached interesting goals on behalf of management of *information as documents*, for example by achieving a complex and comprehensive classification system (presented in section 2.3.).

Even though the advantages of the Linked Data approach are undeniable, and without any desire to entrench our view behind the constituted standards (which, in any case, have already changed), it must be noted that in many cases the data managed on the web are important not for the information contained in them, but rather for the speech act (the corresponding material document) they represent. For example, a speech given by a(ny) President may have a relevance which transcends its content, and is related to the time, circumstances, and events precedent or successive to the speech itself. Those properties can be referred only to that specific act, and the importance of this link is so strong that any extraction of chunks of information provides pieces of data whose total value is way inferior to that of the act as a whole. In the legal field, this is even truer: It is irrelevant to collect a large mass of data if, in order to extract it and put it together, the link between those contents and their juridical sources is weakened.

For these reasons, the approach to legal knowledge representation should study and exploit the arising technologies for the management of information on a content-basis, but at the same time maintain the high standards achieved in document classification.

5.3.1. An approach to the knowledge acquisition bottleneck

The problem of the knowledge acquisition (for a presentation of the issue see 3.5.1.) for these structured data is central in the Linked Data approach, to such an extent that it was faced by [Berners Lee 2009] while presenting Linked Data to the world. In the present section some possible approaches to the issue besides NLP techniques are poited out, starting from the crowdsourcing argument of Berners-Lee himself:

> "Linked Data is all about […] people doing their bit to produce a little bit, and it all connecting. That's how Linked Data works. You do your bit. Everybody else does theirs. You may not have lots of data which you have yourself to put on there but you know to demand it. And we've practiced that". [Berners-Lee 2009]

In the legal field, crowdsourcing could seem difficult to approach, yet it has the potentials to really overcome the knowledge acquisition problem. It is extremely time-consuming to enrich legal texts with structured data, and in order to get an accurate result the user marking up the legal text must be not just a legal expert, but also a specialist in that legal field. Even at that point, the markup may not be complete: annotating some legal text always implies an interpretation of that text, and also case-law from famous courts brings on explicitly contradictory interpretations of the law. It may then be necessary to model different interpretations of a piece of text, and to model the relations between these interpretations. How can all these structured data be gathered?

Lately, the professionals of legal matters spend a lot of their time working in front of a computer, using software such as word editors and law collections. There lays the solution to the issue: to allow multiple, semi-automatic annotation of legal text through a tool implementing a word editor and an interface which allows to compare the user's interpretation with those of other professionals, using a social structure to rate and polish the structured data. The most complete, shared, and useful data will emerge and the legal texts will be able to feature a multiple markup of different views on the document, which would take indefinite time to achieve with a focused data insertion by a restricted group of experts.

The ideal users (and content creators) for such a social platform are the lawyers and the judges. In order for a product such as this to penetrate the category of lawyers (often reluctant to change) it should prove better than commercial legal software for case-law retrieval. This could prove no huge challenge, since this software usually only allow basic search plus some structured data only at the document level (i.e. the classification of the decision depending on their subject, the issuing authority, etc.). Providing to the lawyers a tool which allows them to look for specific (combinations of) legal categories or concepts would greatly improve the number of useful hits while at the same time reducing the time needed to perform those searches. Judges, on the other hand, are public workers: to interpret and explain the law is their job, and the enhancement of their decisions through structured data could be set as a duty for their office. This would represent an enrichment of the case-law directly from the source, which has two positive aspects: first, it provides a basic set of structured data for every (newly issued) judicial decision, therefore providing an outline of its contents. Second, this basic data would come from the most authoritative agent in the process: the judge. All other data, added successively, would acquire a particular meaning when put in comparison to the "original" data (for example, a dissenting interpretation, or a note on successive decisions citing this one).

5.4. Modelling advanced legal concepts

5.4.1. Introduction

The contents of a legal act are very complex objects to model, not only for the issues presented above (logics to be used, strong inheritance from the source document, specificity and ambiguity of language, etc.). Juridical concepts are (quite) ambiguously expressed by legal texts because they deal with a fictional world (that of legal norms) which is itself very complex: it is made of several different layers, influenced by different sources in different ways, and – most complex-increasing element of all – it adapts to circumstances. In fact legal rules, which should (in theory) express strict (general, abstract) relations between behaviours and sanctions, in the practice are never applied straightforwardly. For example:

- Legal rules are subject to **exceptions**, which are not exhaustively listed in a single location;

- Legal rules are **influenced** by other legal rules, and this influence may be not evident from the singular analysis of either of the two rules;

- The interpretation of legal rules is not free, but rather bound to authoritative **precedents** which provide the criteria for the subsumption of facts under legal rules. The ways these criteria are followed (and sometimes the criteria themselves) are, however, complex and inconsistent enough to create a gray area where one cannot be sure whether a legal rule applies or not to a specific case;

- The provisions laid down by legal rules apply in a **temporal frame** which is different from the time of existence of its source documents, and even from that of the legal rule itself.

In the present section a key aspect in legal knowledge modelling will be analyzed, namely, the management of *temporal dimensions*. The successive section proceeds with a deeper analysis of some of the aspects listed above, through a sample modelling of a complex legal concept: that of *efficacy suspension* of norms.

5.4.2. Handling change: the temporal dimension

Temporal dimensions, as well as the geospatial parameters, play a fundamental role in legislative interpretation and application. Jurisdiction is usually closely linked to geospatial information (e.g., Danish law states whether or not a Danish statute applies in Greenland; a regional law applies only in its territory): in these cases the temporal dimension determines a norm's efficacy, and consequently also the purview of its application. In the words of [Palmirani et al. 2010], legal theory identifies at least three axes in a norm's temporal domain:

- An **interval of force**, which acts on the legal text or on a text fragment (article, title, paragraph, etc.);

- An **interval of efficacy**, acting on the legal text or on a text fragment;

– An **interval of applicability**, acting on the norms expressed in the legal document rather than on its text.

The interval of force is defined as the period during which a normative document or provision forms part of the normative system. This period may change over time as the document is modified. Thus, the document's interval of force may be:

– Extended by **prorogation**;

– Brought forward by **anticipation**;

– Struck out by **annulment**, as when the document is found to be unconstitutional;

– Reinstated or **revived**, following abrogation (the interim period between abrogation and renewal being a "gap" in the legal system).

The document is said to be efficacious, or into operation, when it "expresses its normative efficacy" or, better yet, when it may or must be applied. A document's period of efficacy will coincide most of the time with its period of force, except when the period of efficacy is:

– Anticipated, thus beginning before its entry into force (**retroactivity**);

– Extended, thus making it last longer than its period of force (**ultra-activity**);

– Suspended, by way of a **suspension**;

– Postponed, so that it begins after the time of its entry into force (**postponement**);

– Stretched by **prorogation**.

A norm's interval of applicability is the period during which the norm is applied in the concrete, thus producing the effects set out in its provisions. Application normally coincides with efficacy of the document expressing the norms, but the two sometimes differ. The date on which a norm has to be applied guides the judge in applying the norm as part of the case law.

Moreover, such a temporal model should be assigned independently to different parts of the rule such as the antecedent and the consequent (the head and body of the rule) , while at the same time maintaining isomorphism between the legal resource and the formalized rule, following the principle stated by [Bench-Capon and Gordon 2009].

5.4.2.1. Managing the lifecycle of a legal norm

Apart from events influencing the efficacy or applicability of a norm, there are many events which affect the relation between form and content of a norm, or its subject matter, during its period of force. For example, the normative expression could be affected by actions of integration, modification or repealing. In [Lima et al. 2008] the effects of those events on the legal document are represented in the FRBR$_{00}$ standard (see 2.4.2.1.) by a new instance of *Individual Work* with its instance of *Self-contained Expression*. The date of creation of those instances is the date when the modifying norm enters into force. The process is explained in Figure 5.1.

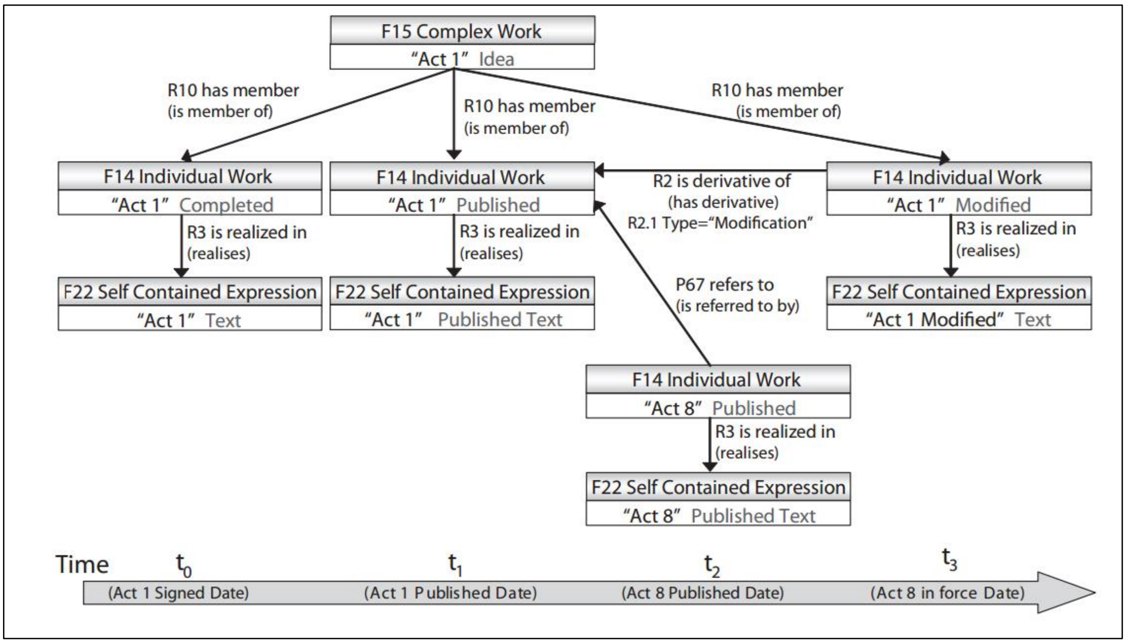


Fig. 5.1 – Dynamics of norms in [Lima et al. 2008]

5.4.2.2. Anatomy of a time-related modificatory provision

Based on this temporal model, which equally applies to the text and to the norms, [Palmirani et al. 2011a] contains a proposal for an anatomy of time-related modificatory provisions:

- **ActiveNorm** (URN). This is a known provision stating a modification;
- **PassiveNorm** (URN, internal/external, complete/incomplete, negative/positive, single/multiple). This is the provision to which a modification applies. The *PassiveNorm* is usually expressed in the text as a normative reference. An *ActiveNorm* and a *PassiveNorm* may collapse in a single document, forming a reflexive modificatory provision, acting on the same text with a self-referring modification;
- **Action** (Type of action, duration of the action, *date_application* of the action, implicit/explicit). This is the type of action the active (or modifying) provision entails for the passive (or modified) one. Actions are organised into a taxonomy, and each action can have a date of application different from the date when the law containing the provision is set to come into force. It is therefore possible to find that a modificatory application has been advanced (brought forward, preponed) or postponed. The action's

duration is a relevant argument for applying the modifications (e.g., a six-month suspension starting on 31 July 2010);

- **TemporalArguments** (describing the start or end of a norm's period of force, efficacy, or application). A modificatory provision is described by three dates: the date of the provision's entry into force, the date on which it becomes efficacious, and the date starting from which it is applied. The goal is to model these temporal arguments so as to help NLP tools detect them automatically;

- **Conditions** (event, space, domain). A temporal modification is sometimes limited to an event, a geographic area, a class (or domain) of application. When a modificatory provision is subordinated to an uncertain event, the action is "frozen" until the condition is resolved. This part of the language is very complex to detect, but the idea is to use a logic formalism to transform these cases into rules with which to logically validate time when the conditions are met. This will determine the time starting from which a modificatory provision will take effect.

Considering that textual modifications contain several hidden temporal modificatory provisions (e.g., substitution of a date), it is important to extract meaningful elements from each modificatory class and to analyze both the legal language expressing each class and the compositional rules or forms used in this language. A degree of regularity in the language and in the expressions used in active modificatory provisions exists, not only for textual modifications but also for exceptions, extensions, and temporal modifications.

5.4.2.3. LKIF-Rule Extension for modelling temporal dimensions

[Palmirani et al. 2011a] presents a modification of the LKIF-Rule language introducing temporal arguments. The extension is made adding the following metadata blocks:

- `events`, listing neutral temporal events;

- `timesInfo`, interpreting the events as time intervals through start/endpoints or duration info;

- `rulesInfo`, connecting the rules with the time parameters.

Following are examples of these blocks and how they are combined to transform neutral events into qualified time intervals. First of all, a list of events is laid down:

```
<events>
    <!-- events of the Order 2007 UK-->
    <event id="e1" value="2007-07-25T01:01:00.0Z"/>
    <!-- Terrorism Act of 2006 -->
    <event id="e2" value="2006-03-03T01:01:00.0Z"/>
    <event id="e3" value="2006-07-17T01:01:00.0Z"/>
```

```
        <event id="e4" value="2006-07-25T01:01:00.0Z"/>

    </events>
```

These events are recalled in the `timesInfo` block:

```
    <timesInfo>

        <!-- Order 2007-->

        <times id="t1">

            <time start="#e1" timeType="efficacy"/>

            <time start="#e1" timeType="inforce"/>

        </times>

        <times id="t2">

            <time start="#e1" duration="P01Y"
    timeType="application"/>

        </times>

        <!-- Terrorism Act 2006 -->

        <times id="t3">

            <time start="#e3" timeType="efficacy"/>

            <time start="#e3" timeType="inforce"/>

        </times>

        <times id="t4">

            <time start="#e4" timeType="application"/>

        </times>

    </timesInfo>
```

These qualified time intervals are recalled in the legal text through the `timesBlock` attribute (see below).

The example in [Palmirani et al. 2011a] explains the modelling possibilities made available by such an extension of a rule language. The example involves the Terrorism Act of UK of year 2000 and its modifications brought by the Terrorism Act of year 2006 and the Order of 2007. Section 25 of the Terrorism Act of 2006 (modifying the detention period set by Terrorism Act of 2000 from 28 days to 14 days) was suspended by the Order of 2007 for the duration of one year. This creates four possible scenarios for the application of the norm:

– The original 28-days detention period, from 2000 to 2006;

– The 14-days detention period between 2006 and 2007;

– The 29-days detention period for 1 year starting on 2007;

– The 14-days detention period from year 2008 on.

The `timesBlock` attribute, assigned to the sentences, defines the temporal parameters of the various parts of the rule.

Following is the representation of section 25 of Terrorism Act of 2006:

```
<rule id="rule3" ruleType="strict" timesBlock="t3">
<!—Sect. 25 Terr. Act/2006 modifies Sch. 8 Terr. Act/2000>
     <head timesBlock="t3">
          <s pred="mod:substituted" id="id4a">
           <v value="sche8_2000">x</v> is modified as if for "28
           days" there were substituted "14 days"</s>
     </head>
     <body>
          <s pred="mod:intoOperation" id="id4b" timesBlock="t3">
          <v value="rule5">y</v> into operation. </s>
     </body>
</rule>
```

Finally, the representation of the Order of 2007, suspending the application of section 25 of the Terrorism Act of 2006.

```
<rule id="rule1" ruleType="strict" timesBlock="t2">
     <!—Disapplication sect. 25 of the Terrorism Act 2006 -->
          <head timesBlock="t2"
               <s pred="mod:suspension" id="id1a">
               <v value="sec25_2006">x</v> is suspended </s>
     </head>
     <body>
          <s pred="mod:enterInForce" id="id3b" timesBlock="t1">
          <v id="sec2_2007">x</v> enters into force </s>
     </body>
```

```
</rule>
```

Through these rules and blocks, the meaning of a suspension switching the days of detention from 14 to 28 in an interval of time starting at `e1` (`2007-07-25T01`) and lasting one year (`P01Y`) is modeled. On the basis of this KB, a reasoner could correctly identify the four different scenarios of application of the norms, through time, and tell which is the maximum duration of detention at a given time.

5.4.3. Modelling the semantics of suspension provisions

The present section presents an example of advanced legal concept modelling. In this example, the concept of *suspension of efficacy* is modeled with a bottom-up approach: the concept is modeled starting from the statutory evidence (a sample of laws where this legal effect is concerned) in order to present a model which could be implemented in a NLP tool, but yet ensures a full support for advanced semantic features.

The aim of [Palmirani et al. 2011b] was to model the suspension of efficacy with FrameNet so as to facilitate automatic detection of arguments in the text, using NLP tools based on a shallow semantic parser. The focus is on suspension because it is more complex and rich with arguments than other temporal modifications, and because it is often used as a legislative drafting technique for introducing a temporary law[25]. The need for temporary laws arises for two main reasons: when the topic is complex and yet urgent, being therefore necessary to find a temporary solution, and when the society needs time to fully apply the new dispositions. *Suspension* may be defined as the action through which a textual provision interrupts the efficacy of a legal text (or fragment thereof) for a given period [Guastini 1998]. In the present model, the text and the actor are one and the same thing: while other authors [Boer et al. 2009] view modifications as documents expressing an authority's action and role, this is a simpler model based on a ontology of documents rather than on a model of the legal system's actors and institutions. Therefore, in this model, all the norms (or rules) contained in a text affected by a suspension are deemed inefficacious without giving account for the authors and the issuing authorities.

Suspension is based on the rationale that some norms so strongly affect their addressees (citizens, businesses, social actors) that an adequate period is needed for them to tune into the process. It is important to clearly recognize, identify and distinguish this rationale over the time even if suspension may come by a variety of different modifications: in other words, it is important to track the entire suspension process even if it is fragmented across several intervals of efficacy, because each macro-suspension is driven by a normative principle. Therefore, the aim is to capture not only a suspension's arguments but also its unity, so as to bring the suspension into relation with its underlying normative rationale.

5.4.3.1. Characteristics of the suspension provision

[25] See The Arizona Legislative Bill Drafting Manual 2009, pag. 6; Texas Legislative Council Drafting Manual, pag. 58, and other US Legislative Drafting Manual.

a suspension provision can be either *explicit* or *implicit*, depending on the language of the provision in question. Under a temporal point of view, an explicit suspension can be either *defined* or *undefined*:

- A suspension is **defined** when its period—the period during which a norm's efficacy is interrupted—is explicitly stated in the text, with the suspending provision clearly indicating a beginning and an end (or an initiating and an ending event). Because the interval is unambiguously defined, it need not be interpreted;
- A suspension is **undefined** when the interval during which a norm's efficacy is interrupted is not explicitly set out in the suspending provision. In those cases, the text often needs to be interpreted to detect the correct value. This class of suspension includes at least three subclasses:
- *sine die* suspension (without an ending date);
- suspension *subordinated to an external event* (e.g., "Article 5 is suspended for a six-month period starting from entry into force of the Treaty");
- suspension *intervals* described with a set of other parameters such as the duration (e.g., "Article 5 is suspended for four months starting from 31 December 2010).

One mode of introducing an implicit suspension is by *disapplication*, which is brought about by a secondary or local law acting through the subsidiarity principle. Disapplication takes also place when resolving conflicts of laws between regional and national law or between national law and European regulations. When a document *disapplies* another document, the latter is frozen, its efficacy being suspended. If the disapplying document is repealed, however, the suspended document is restored to its former efficacy in order to avoid creating a gap in the legal system. This particular mode of suspension is difficult to detect without contextual information, such as the issuing authority, the level on which the law was issued, and the rules framing the legal system.

Another important case to consider is that of a suspension provision being modified. A suspension is usually reflexive, with the law introducing the suspension being the same as that affected by it (this is a role usually devoted in Italy to a law's closing articles). However, it is not unusual to see a later provision modifying the suspension for the same reasons that led to its introduction. For example, Decision 2000/185/EC says under Article 3 that the decision itself "shall apply from 1 January 2000 to 31 December 2002," thus limiting the document's efficacy. Later, Decision 2002/954/EC modified the second subparagraph of Article 3 by replacing "31 December 2002" with "31 December 2003." Then, finally, a third Directive again changed the term, from "2003" to "2005". The rationale guiding this suspension remains the same, and it is important to capture this by first detecting the arguments characterizing the suspension's modification—so as to identify and adjust the main suspension of efficacy—and then describing the phenomena in their atomicity. Note in this specific case that the language of the provision describes efficacy under an inclusion principle and creates two intervals of suspension (one running from the date of entry into force to 1 January 2000, and the other lasting indefinitely – *sine die* – from the second event).

5.4.3.2. Language Regularity of Suspension Provision

In order to model, and consequently extract, semantics from laws introducing or modifying a suspension of efficacy, [Palmirani et al. 2011b] has surveyed a large body of norms that legal practitioners have semantically annotated with Norma-Editor on the basis of the NormeInRete XML schema definition DTDv2.0. The collection includes about 29,000 documents dating from 2005 to 2009, all of them published in Italy's *Official National Gazette* (issued by the country's High Court of Cassation) and selected on the basis of a project on turning these documents into an XML format which CIRSFID worked on over the last five years. On this body of documents a linguistic analysis was performed, in order to isolate patterns for each type of suspension provision. The articles processed are 46,483, and the total modifications are 95, representing 0.2% of the articles. The suspending documents are 90, representing 0.3% of all the documents processed.

The language of suspension has a sufficient degree of regularity to fill the gap between the legal lexicon and the rules of suspension: the provisions always express their temporal arguments and coordinates by way of some temporal expression (an adverb, conjunction, or preposition indicating a continuing, definite, or indefinite time), and they always include a (direct or indirect) reference to the norm whose efficacy is being suspended (*PassiveNorm*).

The logical structure of the suspension norm is *PassiveNorm* is suspended in *TemporalArguments*. Ten terms directly evoke a suspension of efficacy in the document sample. All these terms suspend efficacy, but not all in the same way. More to the point:

- Ten terms (*efficacia*, *efficace*, *applicarsi*, *valido*, *validità*, *effetto*, *applicazione*, *vigore*, *concernere*, and *durata*) introduce an **inclusion of efficacy**, which means that the PassiveNorm will be applied inside the TemporalArguments ("Law X is applied from 1 January 2004 until 31 December 2005");

- Two terms (*sospendere* and *disapplicare*) introduce an **exclusion of efficacy**, which means that the PassiveNorm will *not* be applied inside the TemporalArguments. In other words, it will be applied outside these arguments ("Law X is suspended from 1 January 2004 until 31 December 2005").

The negation of an inclusive form makes the form exclusive, and the same happens with the verb *cessare* (cease) (Law X *ceases* to be *efficacious* on 1 January 2004).

In the logical structure presented above, two elements accompany a suspension-evoking element. They are *PassiveNorm*, representing the norm whose efficacy will be suspended for a certain period, and *TemporalArguments*, time expressions (or time markers) defining the time at which a modification of efficacy is to take effect.

5.4.3.3. Modelling Suspensions Using FrameNet

FrameNet—a lexicon-building project developed at Berkeley University and presented in [Baker et al. 1998] —examines words by their meaning and describes the conceptual structures of sentences. This makes it possible to map the main parts of speech (verbs, nouns, etc.) and to couple them with the legal concepts expressed by the words in question (concepts such as suspension, modified suspension, and disapplication). In order to turn the previously presented elements into Frame Elements within the FrameNet project, it is necessary to model two layers of semantics.

Two frames will be created on the first layer, namely, the **Efficacy_Inclusion** and **Efficacy_Exclusion** frames, which on the second layer will be merged into the *Main_Suspension* frame expressing its meaning in terms of lack of efficacy.

frame (Efficacy_Inclusion) frame_elements (Passive_Norm, Period_Start, Period_End) scene (Passive_Norm has efficacy from Period_Start to Period_End)
frame (Efficacy_Exclusion) frame_elements (Passive_Norm, Period_Start, Period_End) scenes (Passive_Norm is suspended from Period_Start to Period_End, Passive_Norm has not efficacy from Period_Start to Period_End, Passive_Norm has efficacy until Period_End)
frame (Main_Suspension) frame_elements (Passive_Norm, Suspension_Start, Suspension_End) scenes (Passive_Norm is suspended from Suspension_Start to Suspension_End, Passive_Norm has efficacy from Suspension_End to Suspension_Start)

Fig. 5.2 – Frames representing suspension of efficacy.

Efficacy_Inclusion (efficacia.n, efficace.adj, applicarsi.v, valido.adj, validità.n, effetto.n, applicazione.n, vigore.n)
Efficacy_Exclusion (sospendere.v, disapplicare.v, cessare.v+efficacia.n, non.adv+Efficacy_Inclusion)

Fig. 5.3 – Relevant terms evoking the *Efficacy_Inclusion* and the *Efficacy_Exclusion* frame.

The *TemporalArguments* of the shift in efficacy is captured by the *Period_Start* and *Period_end* Frame Elements (FEs), and the target norm is marked as *Passive_Norm*. Frame Element Groups (FEGs) represent the occurrence of FEs in the examined provisions (P=*Passive_Norm*, S=*Period_Start*, E=*Period_End*).

FEG	Annotated Example	Frequency
P, S, E	[P L'obbligo di cui all'articolo 51, comma 1, della legge 27 dicembre 2002, n. 289], **è sospeso** [S dalla data di entrata in vigore del presente	8

	decreto] [E fino al 31 dicembre 2006].	
P, E	[P Le disposizioni del presente provvedimento] **hanno efficacia** [E sino a tutto il 7 maggio 2007].	32
P, S	[P Le disposizioni del presente provvedimento] **cessano di avere efficacia** [S il giorno successivo alle votazioni di ballottaggio di cui al comma 1].	30
P, S+E	[P Le disposizioni della legge 29 dicembre 1988 n. 554] **si applicano** [S+E negli anni 1989 e 1990]	18
S+E, P	[S+E Per l'anno 2008] **non si applicano** [P le disposizioni di cui all'articolo 1, commi 648 e 651, della legge 27 dicembre 2006 n. 206]	3
S, P	[S A partire dal quindicesimo giorno dalla data della pubblicazione della presente legge] **cesserà l'efficacia** [P del decreto ministeriale 22 agosto 1994, n. 582]	4

Fig. 5.4 – Frequency of Frame Element Groups for temporal arguments in the examined sample.

The **Main_Suspension** frame is modelled by inheriting the *Process* frame. Suspension is therefore treated as a process, with a "target" represented by the *Passive_Norm* (carried over unchanged from the first layer) and whose state is affected by one or more events: it starts with the event *Suspension_Start* event and/or ends with the *Suspension_End* event.

FEG	Annotated Example	Frequenc y
P, S	[P Le disposizioni del presente provvedimento] **hanno efficacia** [S sino a tutto il 7 maggio 2007].	48
S, P	[S A partire dal quindicesimo giorno dalla data della pubblicazione della presente legge] **cesserà l'efficacia** [P del decreto ministeriale 22 agosto 1994, n. 582]	4
P, S, E	[P L'obbligo di cui all'articolo 51 della legge 27 dicembre 2002, n. 289], **è sospeso** [S dalla data di entrata in vigore del presente decreto] [E fino al 31 dicembre 2006].	2
P, E, S	[P Il presente provvedimento] **acquista efficacia** [E dalla data di indizione dei comizi elettorali per i referendum regionali] [S sino a tutto il 5 ottobre 2008]	10

S+E, P	[S+E Per l'anno 2008] **non si applicano** [P le disposizioni di cui all'articolo 1, commi 648 e 651, della legge 27 dicembre 2006 n. 206]	3
P, E	[P L'applicazione dei commi 1 e 8 dell'articolo 177] **resta sospesa** [E fino alla data di entrata in vigore del regolamento di cui al comma 5]	13
P, E+S	[P Le disposizioni della legge 29 dicembre 1988 n. 554] **si applicano** [S+E negli anni 1989 e 1990]	15

Fig. 5.5. Frequency of Frame Element Groups in the main suspension frame.

The following table explains how the *Period_Start* and *Period_End* elements are transformed into *Suspension_start* and *Suspension_end*, marking respectively the beginning and end of a period where a norm lacks efficacy.

FE to be trasformed / Evoked frame	**Efficacy_Inclusion**	**Efficacy_Exclusion**
Period_Start	*Suspension_End*	*Suspension_Start*
Period_Stop	*Suspension_Start*	*Suspension_End*

Fig. 5.6. – Conversion of period markers into suspension markers.

Also, a **Suspension_Modification** frame will be created to capture provisions modifying a suspension previously introduced by another norm. It is quite easy to distinguish between the two kinds of provisions, since they are textual modifications lacking a term that evokes an *efficacy* frame and contains some *Change_event_time* frame. In order to properly interpret the modification, there needs be a comparison between the *Suspension_Modification* and the *Main_Suspension* (contained elsewhere). For this reason, the *Suspension_Modification* element will be presented in FrameNet without any semantic specification of its content, since the exact interpretation of the provision will be entrusted to other tools.

frame (Suspension_Modification) inherits from (Change_Event_Time) frame_elements (Passive_Norm (=Event), Suspending_Expression (=Event), Suspension_Start (=Landmark_Time), Suspension_End (=Landmark_Time), Undefined_Suspension_Modification (=Destination_Time)) scene (Passive_Norm Suspending_Expression is postponed from Suspension_End to Undefined_Suspension_Modification)

Fig. 5.7 – The suspension modification frame.

Moreover, the start of the process can be advanced or postponed by another norm, and the same can happen to its end. These events will be represented in four specific frames, subclasses of the *Suspension_Modification* frame.

5.4.3.4. From the Text Structure to the Legal Semantics

The FrameNet model described above is designed for automatic detection of the provisions (and relative arguments) by which a norm is suspended, with the help of NLP tools. On the structural level, a parser automatically detects *references*, *dates*, and other metadata in a legislative text and translates this information into XML tags and metadata using legal XML standards (such as Akoma Ntoso and NiR). The XML file is then processed by a deep syntactical parser based on a NLP tool. Detecting the treebank and the main linguistic structural elements can help understand the semantics acquired from a FrameNet analysis of the suspension. Efficacy-evoking terms help understand the type of provision in question: if the evoking word occurs as the subject, then the prepositional phrase is marked as *Passive_Norm* (as in "Efficacy of law X"); if the evoking word occurs as the predicate, the *Passive_Norm* element will be represented by the subject ("Law X is suspended"). Words expressing time are marked as *Period_Start* or *Period_End*.

FrameNet distinguishes between *Efficacy_Inclusion* and *Efficacy_Exclusion* frames, and the frame's *Period_Start* and *Period_Stop* are converted into *Suspension_Start* and *Suspension_End* accordingly. FrameNet also recognises the presence of a nearby *Change_Event_Time* frame and introduces a *Suspension_Modification* frame in place of a *Main_Suspension* (which should already be effective in *Passive_Norm*). In this case, the frame's *Suspension_Start* and *Suspension_End* are replaced with *Start_Modification* and *End_Modification* and a reasoner processes the *Suspension_Modification* frame in light of the *Main_Suspension* contained in the *Passive_Norm*, distinguishing advancement and postponement of start/end events (*Start_Advancement*, *Start_Postponement*, *End_Advancement*, *End_Postponement*).

The regularity of the modificatory provisions and the frequency of the textual ones encourage research in the direction indicated. Moreover, the results obtained from applying NLP tools bear out our methodology: a more detailed vocabulary of verbs can be extracted from a large database, and that would make it possible to reinforce the linguistic classifications in conjunction with a taxonomy of modificatory provision. On the other hand, when dealing with temporal modifications more contextual information is needed, together with an account for the linguistic patterns used for expressing the action, a more powerful instrument for describing regularity, and a connection with the lexicon. It is for this reason that FrameNet may turn out useful.

[Palmirani et al. 2011b] presents a first outcome of this approach, meant to be an advancement when it comes to modelling legal language while taking into account the theory of law. This research showed how important it is to fully understand the legal phenomena and to model them with logical rules, which in turn are supported by the state-of-the-art in language processing. It showed how the concept of suspension requires advanced modelling in order to lay down the necessary set of metadata to fully capture its meaning and its effects: this kind of effort has to be replicated for every legal phenomenon, and at the same time it is necessary to leave space for modification of such models, so that the tool evaluating the legal concept can rely on a custom theory which is the one that the user wants to adopt. As already noticed in this chapter, such tools assume more and more the appearances of the ancestral tools called *legal philosophy* and *legal doctrine*.

5.5. Different perspectives on representing judicial knowledge

5.5.1. Introduction

The present section contains a comparison of the main models used in the state-of-the-art of legal knowledge representation and reasoning. Those approaches (and most of the solutions they embody) have already been presented in this thesis, but in this section they will be analysed together in one of their applications. The purpose is to understand the differences between these approaches, and to compare the features presented by those authors with those indicated so far, seeing if they deal with the same issues pointed out in the present chapter, and how effectively they do it.

Most of the research in the legal subjects uses some famous examples as a test field for theories and applications. In case-law, the example used as a paragon is the American case of *Popov v. Hayashi*[26], decided by the honourable Kevin M McCarthy, a pioneer in AI & Law (see 1.4.1.). The case concerned the possession of a baseball hit by Barry Bonds for his record breaking 73rd home run in the 2001 season. Such a ball is very valuable (Mark McGwire's 1998 70th home run ball sold at auction for $3,000,000). When the ball reached the crowd, Popov caught it in the upper part of the webbing of his baseball glove, in a catch which does not give certainty of retaining control of the ball, particularly since Popov was stretching and could have fallen. However, Popov was not given the chance to complete his catch: he was tackled and thrown to the ground by others trying to secure the ball, which fell off his glove. Hayashi, himself innocent of the attack on Popov, picked up the ball and put it in his pocket, so securing possession.

5.5.2. A legal ontology to represent the case

[Wyner and Hoekstra 2010] presents an OWL ontology called *Legal Core Ontology*, to make explicit the conceptual knowledge of legal case domain, support reasoning on the domain, and annotate the text of cases.

The ontology is intended to be open to additions (i.e. with ontologies representing domain-specific knowledge) focusing only on legal concepts which are strictly related to case-law: it is broadly compatible with [Costa et al. 1998]. It also leaves concepts such as events/processes, causation, time, space, propositional attitudes, and so on to external extensions.

Another important design choice concerns the representation of extra-legal or upper domain knowledge: it is represented in the Legal Case Ontology only as far as legal representation on

[26] Popov v. Hayashi, 2002 WL 31833731 (Cal.Superior Dec 18, 2002) (NO. 400545).

the matter differs from the general one (for example, a tomato is considered a fruit under a biological ontology, a vegetable under a legal one).

Following this approach, and similarly to the ontology created during the present Ph.D research, the ontology does not represent the *dynamics* of reasoning, but the *static* data over which participants argue: i.e. it assumes that some cause of action holds in the case, without representing how this comes to be.

The central concept of the ontology is the *case* class, which is connected to the other main concepts *Decision*, *Jurisdiction*, *Participant*, *Argument Scheme*, *Element*, *Evidence*, *Legal Concept*, *Legal Document*. Among those, Argument Scheme and Legal Concept are particularly interesting, as they introduce *argument schemes* and *factors* into the case ontology.

5.5.3. Bench-Capon: introducing dimensions and factors

The reconstruction in [Bench-Capon 2012] goes beyond the mere classification of concepts, but does not delve into pure formal reasoning either. Here, the case is dealt with logics, and the argumentation is built upon the theory construction style of [Bench-Capon and Sartor 2003]. Following this methodology, a set of factors is identified together with the value they promote and the side they favour (this is done by starting with a dimensional analysis and using the various points on these dimensions as factors, just like Cato derived factors from the dimensions of Hypo). The factors are then assigned to a set of cases (*precedents*) which have to be compared to Popov vs. Hayashi, and their outcome is recorded.

As far as logics is concerned, the analysis highlighted the need for defeaters in the sense of Defeasible Logic. However, the author also notes that only some of the factors behave as defeaters (rules blocking a conclusion while not licensing the negation of that conclusion), and so factors can no longer be seen as homogeneous and do not straightforwardly relate to rules. This result seems to confirm the choice made for IBP in [Brüninghaus and Ashley, 2005], where a distinction between factors was found to be required in order to be able to resolve conflicts appropriately.

Also argument schemes are used to represent the case, but this is achieved at the cost of renouncing to dimension (and the different degrees of support they allow): for this reason, factors are split into a pro-plaintiff and a pro-defendant version.

5.5.4. Evaluating arguments with values

Values, together with defeasible logics, were first used to model the arguments of Popov v Hayashi in [Wyner et al. 2007]. This reconstruction was based on prior works on valued-based argumentation systems [Bench-Capon and Sartor 2003] and on modelling practical reasoning [Atkinson 2005]. It used several argumentation schemes, including *arguments from witness testimony* and *defeasible modus ponens*. However the single arguments were not modelled. The analysis of the decision is done through a set of defeasible rules which allow intermediate conclusions to be derived from the facts. Arguments are built from these intermediate conclusions and attack relations are asserted between them, building a Dung abstract argumentation framework. The last part of the decision (the equitable solution) is modelled using a value-based argumentation framework. The whole reconstruction was done manually, without support from a software tool for their formal methods.

5.5.5. Prakken's argumentation framework

The framework of [Prakken 2010] is applied in a practical representation of the *Popov vs. Hayashi* case in [Prakken 2012]. Here, the concepts of defeasible rules and Dung semantics constitute the semantics for argument acceptability.

The representation of the famous case allows the author to apply one of the pillars of his framework, namely the concept of *reinstatement*: an argument with defeaters is still justified if all its defeaters are in turn defeated by a justified argument.

The two technical solutions used to represent the case (and successfully evaluate the arguments in accordance with judge McCarthy's decision) were rule-exception structures and arguments about the validity of rules, which turned out to be sufficient to capture value-based reasoning, since, as the author notices, "the only value that was at stake in this case, fairness, was reconstructed as a condition of a validity rule".

Another peculiarity of the experiment was the concept of two-steps argument: the passage from the premise to the final conclusion is mediated by an "intermediate conclusion". This better captures the dependency relations between the elements of arguments, since complex arguments are usually not expressed with the classic attack-counterattack relations. In accordance with Gordon's view on attacking arguments, the argument against the final conclusion is seen as a separate kind of speech act: a challenge of the conclusion, which can in turn be replied with a further argument for that premise.

This introduces dialogical aspects into the argumentation modelling, bringing this model of the dynamics of the *Popov vs. Hayashi* case very close to the goal of the present research in the representation of the judicial reasoning.

5.5.6. Reconstructing the case with some Carneades tools

The Carneades Argumentation System has been tested in the *Popov vs. Hayashi* case in [Gordon and Walton 2012]. The paper illustrates Carneades' support for argument reconstruction, evaluation and visualization. However, not all features of Carneades were used in that demonstration: argument construction from knowledge base, in particular, was not used as the knowledge of the domain would have needed to be modelled at a finer level of granularity.

5.5.7. A comparison of the argument models for Popov vs. Hayashi presented so far

5.5.7.1. Tasks

The purposes of the OWL ontology in [Wyner and Hoekstra 2010] are quite similar to those of the ontology set of the present research: to make explicit the implicit knowledge of legal cases that legal professionals have, and to build a database of cases which applies automated case-based reasoning rules to answer queries on the knowledge base, even over the internet with a web-based tool. When dealing with the choice of lexicon, the authors decide to set aside the existing standards, which are considered too general, and build a language from the scratch following [Ashley 2009] which sets out three roles for legal case ontologies: to support case-based reasoning, to distinguish between deep and shallow analogies, and to induce and test hypotheses. The task of supporting reasoning is treated only to a certain extent, since it is not intention of the ontology to introduce rules relating cases to each other (which is the heart of case-based reasoning).

The example presented in [Bench-Capon 2012] is an effort to apply the dimensions and factors approach to case based reasoning in AI and Law.

[Wyner et al. 2007] wants to show that useful argument schemes and associated critical questions can be extracted from a complex case, then input into an argumentation framework in order to reach the claim of the case. It provides an example of how a real case can be represented and reasoned with using argument schemes and argumentation frameworks, thus providing an empirical basis for further research and applications of argumentation.

Prakken's example was aimed at testing the suitability of the abstract framework of [Prakken 2010] for modelling the argumentation structure of the decision.

[Gordon and Walton 2012] is focused on reconstructing and critically evaluating Judge McCarty's arguments in his decision, similar to what has previously been made in [Gordon and Walton 2006].

5.5.7.2. Methodology

The methodology chosen for the OWL ontology presented in [Wyner and Hoekstra 2010] is very interesting for the purposes of the present research: in fact, instead of trying to *minimise* reasoning by highly specifying the ontology (i.e. by explicitly asserting all subclasses and property relations), the approach followed attempts to *maximise* reasoning (i.e. deriving classes and subclasses from class restrictions and axioms): this allows individuals to vary their properties depending on the context in which they appear, which is exactly what an ontology representing case-law needs (i.e., a person can be the plaintiff in one case, a defendant in another case; a certain characteristic of a contract can be judged as sufficient for the adjudication in one case, and not sufficient in another one).

In [Bench-Capon 2012], in order to apply the dimensions and factors approach to Popov v. Hayashi, first a suitable group of precedent cases is assembled. These are:

- *Keeble v. Hickergill*, where the defendant, out of malice, scared ducks away from the plaintiff's pond;
- *Pierson v. Post*, where the plaintiff intercepted and killed a fox which the defendant was hunting;
- *Young v. Hitchens*, where, while fishing, the defendant spread a net inside the plaintiff's bigger net, catching the trapped fish;
- *Ghen v. Rich*, where the defendant bought a whale washed ashore after having been harpooned by the plaintiff.

These cases are then represented first using factors and then dimensions, which allow a finer grained, and hence more satisfactory, representation of the issues involved. The cases and the various factors are, in this way, rearranged in several ways in order to highlight the complex network of common points and differences. Next, Hypo-style three ply-arguments are constructed: in turns, the plaintiff and the defendant cite favorable precedents, explaining their analogies with the current case in order to win the claim, and try to distinguish adversarial citations.

The example provided in [Wyner et al. 2007] proceeds in four phases:

- It is established what **facts** the court will acecept, based on the presentation of the case;
- From the facts, **rules** are provided enabling to reason from the facts to intermediate terms (some of which are legal terms and some not), that then form the basis of the legal analysis;
- **Arguments**, which are applications of the rules, are expressed abstractly and organized into an abstract argumentation framework which specifies the attack relations between the arguments, from which sets of acceptable arguments can be identified;
- **Values** are associated with the arguments, further refining the argumentation framework. Given an audience, understood as a value-ranking on arguments, the final outcome of the case is calculated.

Prakken's formalization defines arguments as inference trees formed by applying two kinds of inference rules, strict and defeasible rules, as described in 1.3.2.2. as an application of the principles of defeasible logics presented in 1.2.7.).

The methodology used in the reconstruction and evaluation in [Gordon and Walton 2012] is summarized as follows:

- All **statements** (atomic propositions) of arguments are listed, and each is given a identifier;
- **Premises** and **conclusions** of arguments are identified among the statements;
- The Carneades Mapping Tool is used to create the **argument diagram**, linking shared statements of arguments;
- Each statement is given a **proof standard**;
- Each argument is labelled with an **argumentation scheme**;
- Statements which are **assumed** to be true without argument by the *audience* (in this case, the judge) are marked.

At this point, the Carneades Software is capable of using the arguments to reason forwards from the accepted and rejected statements, noting which arguments are applicable because all of their premises hold, and which statements are acceptable because the applicable arguments satisfy the statement's proof standard.

Finally, argumentation schemes can be used to critically evaluate the arguments, revealing implicit premises and critical questions.

5.5.7.3. Implementation of Factors, Dimensions, and Values

In [Wyner and Hoekstra 2010] values are considered only to the extent of the cause of action (the claim) they sustain, while dimensions and factors are merged in the OWL ontology in a subclass of the *legal concept* class called factor. The authors notice that these factors should have a hierarchical organization, where basic level factors relate to higher level, more abstract factors. This is however not necessarily a structure of strict subsumption so cannot simply be expressed as a class hierarchy. For this reason, factors are not furtherly defined in the ontology (factor-based reasoning being, by the way, beyond its scope). Comparing the case to similar precedents, [Bench-Capon 2012] identifies the following factors:

- F1 **Not-Caught**: The animal was neither in the bodily possession of the plaintiff, nor mortally wounded. Advances the purpose of legal certainty by providing a clear definition of possession. It is pro-defendant;

- F2 **Own/Open**: *Own* applies if the plaintiff was hunting on his own land and advances the purpose of protection of property rights. *Open* applies if the plaintiff was hunting on open land. Both are pro-plaintiff;

- F3 **Livelihood**: The plaintiff was engaged in earning his living. The purpose advanced is the protection of valuable activity, and it is pro-plaintiff;

- F4 **Competition**: The defendant was in competition with the plaintiff. This advances the purpose of promoting free enterprise, and is pro-defendant.

It turns out that Popov vs Hayashi presents only F1 and F4, and that no factor favoring Popov is present. A more subtle analysis is needed to do justice to Popov's case, it remains to see if dimensions can supply what is required. The reference work for the role of dimensions is [Bench-Capon and Rissland 2001]. As well as considering the different degrees of closeness to bodily possession, they suggest that the key importance of land ownership (at least in cases where the question of trespass does not arise) is that the owner of the land may be considered to own the animals in virtue of their presence on his land, without any need to physically seize them. Moreover, the factors relating to livelihood seem rather narrow: they do not allow to take account of the arguably socially useful role of Post's fox hunting, nor of the fact that the defendant in Keeble was acting out of malice. Without this there is no account neither for the strength of Post's case nor of Keeble's, and these strengths do matter in the current case. Bench-Capon hence uses the following four dimensions to represent the cases:

- D1 **Possession**: This would range from the extreme pro-defendant position where the animal was roaming entirely free, through chase being started, hot pursuit, mortal wounding to the extreme pro-plaintiff position of actual bodily possession;

- D2 **Ownership**: This would range from the the extreme pro-defendant position where the animals never entered the plaintiff's property, through various degrees of frequency of presence, from straying to regular and predictable visits, to the extreme pro-plaintiff position where the animal was incapable of leaving the land, where we had a fox on an island, for example;

- D3 **PlaintiffMotive**: Rather than simply focussing on whether the plaintiff was earning his livelihood, the suggestion here is that a range of increasingly worthy motives should be considered, starting from malice, through pleasure and social service to livelihood;
- D4 **DefendantMotive**: As for the plaintiff motive, but relating to the defendant.

This finer grained representation allows to arrange the cases on four dimensions in the following way (naming the most pro-plaintiff case on each dimension first). Note that the Ownership dimension applies only to Keeble.

- *Possession*: Ghen, {Popov, Young}, Keeble, Pierson;
- *Ownership*: Keeble;
- *PlaintiffMotive*: {Ghen,Young, Keeble} Popov, Pierson;
- *DefendantMotive*: Keeble, Pierson, Popov, {Ghen,Young}.

The dimensions discussed so far are all taken from analysis of the wild animal cases, which was undertaken without any thought of Popov. This is, of course, in the spirit of systems such as Hypo and Cato which presuppose a set of cases to which the current case will be presented. Bench-Capon thinks instead about a representation with Popov at the forefront, which is in line with [Levi 1948], holding that past cases are potentially reinterpreted in the process of being applied to new cases. Looking at the section of McCarthy's decision in which he discusses the evidence in order to establish the facts on which he will base his decision, it turns out he has three major concerns:

- He establishes that Popov did not complete his catch, and so he was never in possession of the ball;
- He is able to establish that Popov was prevented from attempting to complete the catch by an illegal action on the part of a group of unidentified persons;
- He establishes that Hayashi was not one of the people involved in the illegal act.

Only the first of these facts relates to a dimension (*possession*), while the other two do not relate to the dimensions identified so far, and therefore Bench-Capon identifies new dimensions which can effectively face the McCarthy's concerns and yet be applicable to the precedents:

- **Possession**: Ghen, Keeble, {Young, Popov}, Pierson;
- **Interference Illegal**: Popov, Keeble, Pierson, Young, Ghen;
- **Defendant Behaviour**: Keeble, Pierson, Young, Ghen, Popov;
- **Activity to Encourage**: Ghen, Keeble, Pierson {Young, Popov}.

As a result, it turns out that the factors based approach aligns the case with Pierson, and the dimensions approach with Young. However, considering the perspective just developed, Popov (and Pierson) has a dimension on which Young can be distinguished: the interference was illegal. Although this was not something that the analyses is previous papers recognised, it was a clear consideration of the judge in the Popov case. Thus Bench-Capon showed practically how a new case (Popov) requires to modify the analysis of precedents (Young) to include an additional dimension. It is not surprising that a new case can lead to a reinterpretation of existing cases. Case law should not be seen as a static body of knowledge, but as something which evolves and adapts. In the words of [Levi 1948]:

> "The movement of common or expert concepts into the law may be followed. The concept is suggested in arguing difference or similarity in a brief, but it wins no approval from the court. The idea achieves standing in society. It is suggested again to a court. The court this time reinterprets the prior case and in so doing adopts the rejected idea." [Levi 1948]

For more on this subject, see 1.3.7.5.

In [Wyner et al. 2007] factors are not considered, while values are introduced into the argument scheme-based reasoning through a specific argument scheme, called *Argument from purpose*, a variant of the scheme for practical reasoning which represents McCarthy's teleological reasoning.

Being fully developed upon formal logics, the models of Prakken and Gordon have small space for factors and values. As for Atkinson, factors are present only as facts (statements) and only in this form could some comparison between different precedents be possible. Values have even less space in the representation of Prakken (only *fundamental fairness* appears in the premises of a cause of action for equity), and Gordon tries to overcome this lack with the *equitable solution principle*, which actually introduces values into McCarty's reasoning. This is how Gordon and Walton achieved the goal of matching the final outcome of *Popov vs. Hayashi* through an automatic application of rules, facts and schemes by the Carneades system.

5.5.7.4. Modelling of arguments

The OWL ontology of [Wyner and Hoekstra 2010] contributes to the modelling of arguments by defining arguments schemes, in particular the *eyewitness* and *expert* schemes. As the authors declare, there are significant issues concerning how much of the semantics of arguments schemes can be expressed as OWL-DL class restrictions. In order to incorporate them, the authors define unique necessary and sufficient conditions for arguments schemes. For instance, the presence of an expert witness testimony indicates that the argument scheme used to provide the rationale for a case was at least of the type. The authors also notice that argument schemes may include exception clauses and often express plausible reasoning pattens rather than certain ones, and are therefore outside the scope of DL. This is the reason why argumentation schemes are used only for classifying cases according to the reasoning patterns used to make a determination. Notably, the authors also propose the modelling of legal theories, legal rules, and causes of action as argument schemes, even if they don't proceed in that direction since they recognize these as complex concepts which merit further consideration and development.

The reconstruction in [Bench-Capon 2012] follows the idea of Cato that an argument based on following a precedent could be presented as a cascade of argumentation schemes. The program starts by comparing the factors present in the current case and a precedent and partitioning them according to whether there are present in both cases or only one. It then uses the following argument schemes:

- **AS1**: the main scheme, which argues that the current case should be decided in the same way as the precedent on the basis of their shared factors. The preference between factors is justified by AS2 but is subject to an exception represented by AS4. This scheme favors the proponent (the plaintiff given the above partitions);
- **AS2**: this states that the shared plaintiff factors were preferred to the shared defendant factors in the precedent. It is subject to an exception represented by AS3. It favors the proponent;
- **AS3**: this says that the precedent was stronger for the plaintiff, on the basis of its factors. If, however, there are factors with the same value in the current case, these may provide a counter argument. This attacks AS2 and so favors the opponent;
- **AS4**: this says that the current case is stronger for the defendant on the basis of factors in a precedent. If, however, there are factors with the same value in another precedent favoring the defendant, these can be used as a counter argument. This attacks AS1 and so favors the opponent;
- **AS5**: this is an additional argument to find for the plaintiff based on strengths in the current case not used in AS3. This provides additional support for the proponent;
- **AS6**: this is an additional argument to find for the plaintiff based on weaknesses in the precedent case favoring the defendant not used in AS4. This also provides additional support for the proponent.

Because dimensions are not used here, the degree of support is not considered. For this purpose therefore a pro-plaintiff and a pro-defendant factor are used, relating to each of the five values.

The arguments reconstruction of [Wyner et al. 2007] gets very close to the state-of-the-art of legal arguments representation, with facts, rules and arguments. First, the authors define a basis of established facts F1-F6, and then build arguments in the form of simple logical rules, starting from the concepts of possessions and introducing the specific rules about baseballs and gloves, control, motion of the object, contact, and so on. These specific rules are, in turn, backed by precedents, or by custom and practice.

In [Prakken 2012], all arguments are built through logic formulas, “from scratch” so to say, starting from the building blocks of argument schemes: the *rule validity scheme* and the *defeasible modus ponens* scheme (see 1.3.4.2.). In this way, the schemes for Witness testimony and Video tape are built. Facts (F1-F9) and three sources of rule validity (V1-V3) are then formalized, before proceeding with the set of rules concerning possession, catching, physical control, qualified possession. Some of these elements are then used to build the *analogical rule validity scheme*. Arguments are then generated from by reinstatement of the rules into an argumentation scheme.

In the example in [Gordon and Walton 2012], arguments are modeled depending on the answers they are meant to resolve. First, the following factual issues are modeled:

Did Popov catch the ball?

Was Hayashi guilty of any wrongdoing?

At that point, arguments schemes are introduced in the form of critical questions, which in turn present themselves in the graph as a special type of premise (exceptions and assumptions). Other graphs are built for issues such as the cause of action, the ownership and the possession of the ball, the backing of the conversion warrant, Popov's right to possess the ball, and finally the need for an equitable solution. Here, facts are not used as a starting point to form complex statements, but the statements are considered to be the main blocks of argumentation. This approach abstracts a bit from the legal elements (and restitutes a structure which brings weaker resemblance to the small factors involved in the case), yet it leaves an open end to the text metadata layer.

5.5.7.5. Modelling of attack relations

As previously said, the OWL ontology does not consider the *dynamic* aspects of argumentation. Attack relations are therefore not modeled, and argument schemes are present only for the classification of the reasoning patterns.

[Bench-Capon 2012] uses argument schemes to build a graph of the decision, which represents the reasoning in the case as a tree of argument schemes.

The root of the argumentation represented in the graph of Figure 5.8 is the claim that the case should be found for the plaintiff. The children of a claim node are the argument schemes which have been instantiated to support or attack it. The children of these scheme nodes are the

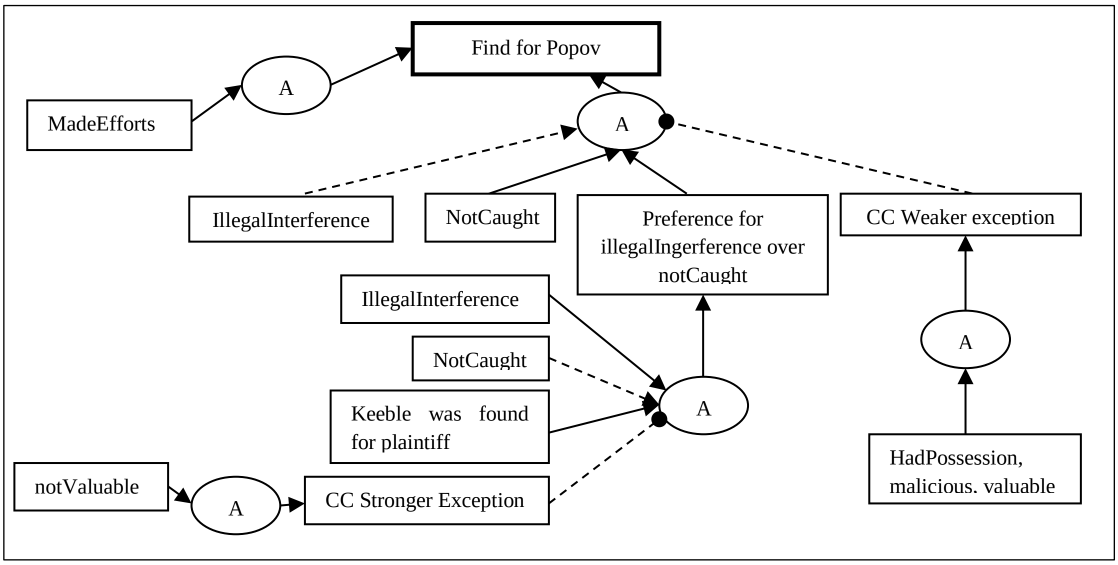


Fig. 5.8 – Graph representing argumentation for Popov in [Bench-Capon 2012].

premises and any exceptions. Premises and exceptions may themselves be claims of further argument schemes. The central argument, which is based on the preference for malicious interference over the fact that the animal was not caught, is attacked by the lack of utility in Popov's activity. But if we prefer Public Order to Utility, we may reject this counter argument. The argument is also problematic because Popov lacks the bad defendant motive, Keeble's valuable activity, and Keeble's possession claim based on land ownership. In order to reject this, we must prefer Public Order to all the three values represented by these factors, even in combination. Finally we have an additional argument for Popov in that his efforts might deserve some reward. A similar graph can be constructed to show the case for the defendant based on Young, as in Figure 5.9.

Here the validity of the argument turns on whether we give sufficient weight to Public Order to block the preference using either AS3 or AS4, but even if we do, there are arguments available based on the valuable nature of Young's activity against the selfish gains sought by Popov. Taken together the two graphs imply that Public Order must be accorded supreme importance if Popov's case is to stand, and even this may not be sufficient to find for Popov. Bench-Capon

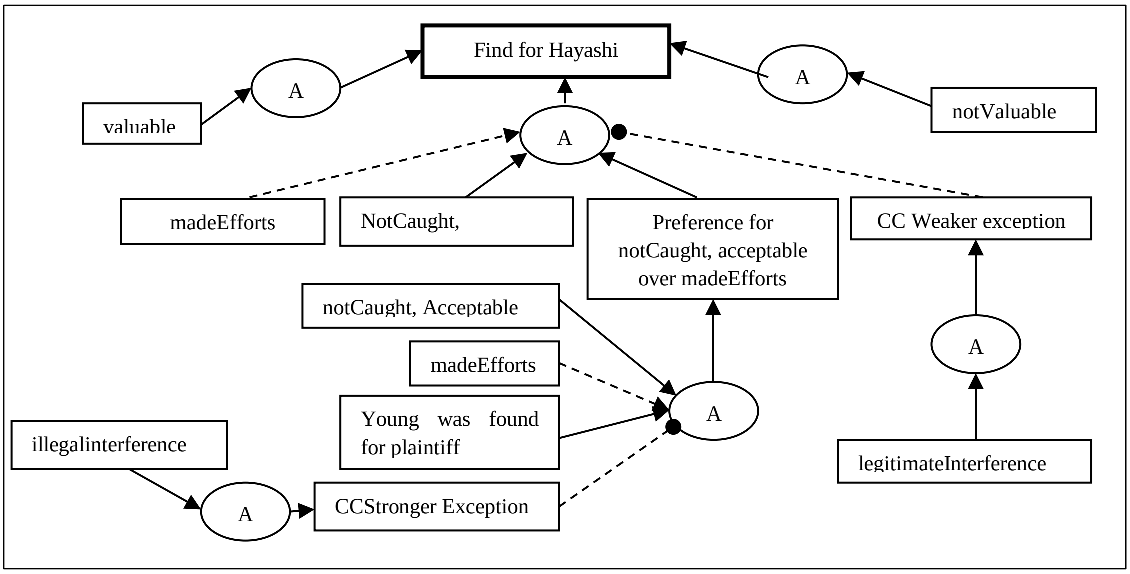


Fig. 5.9 – Graph representing argumentation for Hayashi in [Bench-Capon 2012].

concludes that "Public Order might be given this high importance if it were desired to send a clear message that impeding people attempting to catch valuable balls would be futile, since that person would be awarded the ball even if it were recovered by someone who himself did no wrong".

In [Wyner et al. 2007] arguments are evaluated in a Dungean argumentation framework. An argument *An*, for some *n*, is an application of rule *Rn*, for that *n*. In addition to arguments based on the rules, four additional arguments are introduced, labeled with questions, intended to represent the inference on the basis of arguments which, unlike video and witness testimony, are not observable. A graph is built, showing that three arguments have to be accepted. In order to consider the arguments from purpose, a set of practical reasoning arguments are built and a parallel value-based graph is built:

- PR1: Where interruption of completing the catch so establishing possession was illegal: decides for Popov, prevents assault being rewarded, promotes the value of public order;

- PR2: Where it has not been shown that Hayashi did not have possession and did nothing wrong: does not decide for Popov, would punish Hayashi, demotes the value of fairness;

- PR3: Where Hayashi had unequivocal control of the baseball: decides for Hayashi, provides a bright line, promotes clarity of law;

- PR4: Where interruption of completing the catch so establishing possession was illegal; do not insist on unequivocal control; which would reward assault; demoting the value of public order;

- PR5: Since Hayashi was not an assailant, finding for Hayashi would not reward assault;

- PR7: Where interruption of completing the catch so establishing possession was illegal: Popov should sue the assailants of the assault, would not punish Hayashi, promotes the value of fairness;

- PR8: Since assailants cannot be identified, suing those responsible for the assault is not a viable action.

The practical reasoning arguments are presented in Figure 5.10. In order to evaluate the justifiable arguments, it is necessary to identify the value ranking first. McCarthy has explicitly said that fairness will be his primary value. Thus, both PR1 and PR3 are defeated. PR5 will defeat PR4, and PR8 will defeat PR7, since both must be objectively accepted given the facts. This leaves only PR2 and PR6. Note that these are not in conflict: the result is that McCarthy can decide neither for Popov nor for Hayashi. In fact, he decided that the ball should be sold and the proceeds divided between the two.

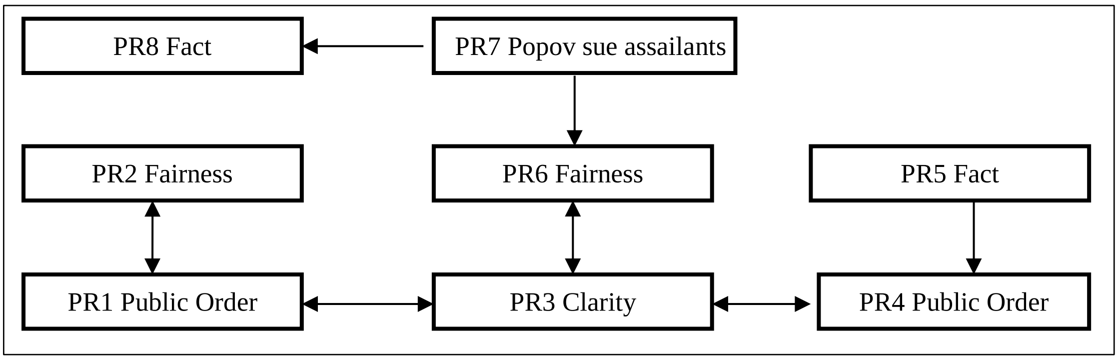


Fig. 5.10 – Graph representing arguments in [Wyner et al. 2007].

Attacking relations in [Prakken 2012] follow an argumentation theory based on dungean semantics which distinguishes between justified, overruled and defensible arguments. The final conclusion of its application to the argument graphs is that both Popov and Hayashi should win. Judge McCarthy did not resolve these conflicts with preferences, so that both rebuttals result in symmetric defeat relations. Instead, McCarthy defeated the two rebuttals by premise-attacking them on their assumptions, resulting in asymmetric premise-defeat and since the two defeaters themselves have no defeaters, they are both justified. MC is the rule found by McCarty to decide the case:

x has a legitimate claim to property *y and x* is plaintiff in a conversion case *and*

z has a legitimate claim to property *y and z* is defendant in a conversion case *and*

the claims of *x* and *z* are equally legitimate *and*

nobody else has a legitimate claim to *y then presumably*

x and *z* must equally share the value of property *y*

The scheme can be extended with argumentation for each of the mutually rebutting arguments, as well as for the grounds of validity of rule MC. Prakken also gives account for the allocation of burden of proof. In the present case it was uncontroversial that Popov had the burden of persuasion for his main claim with a proof standard 'preponderance of evidence' and that there were no shifts in the burden of persuasion for sub-issues. As suggested by [Prakken and Sartor 2009], this can be modelled in the present framework by requiring that Popov wins if and only if a justified argument for his main claim can be constructed. In his work, Prakken applies proof standards by deciding whether an undercutter or invalidity argument must be moved against an evidential argument: the lower the proof standard, the easier such a counterargument will be moved.

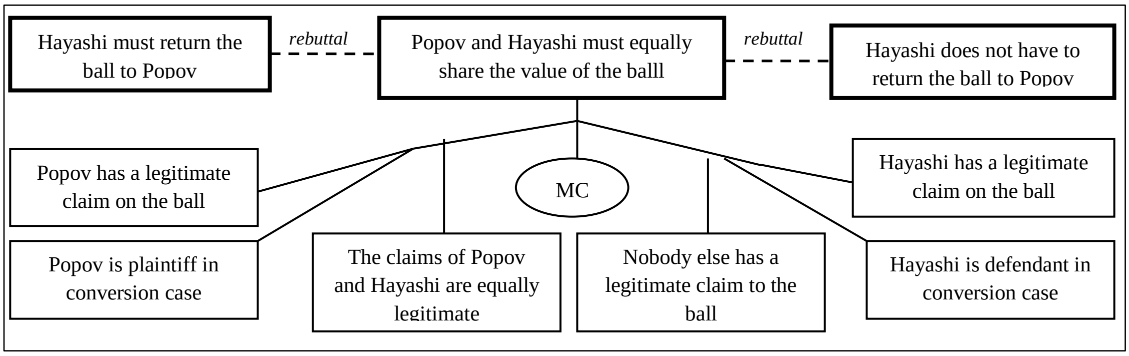


Fig. 5.11 – Graph representing the final conclusion in [Prakken 2012].

In [Gordon and Walton 2012] arguments are evaluated following the argumentation theory described in Chapter 3. The evaluation is based on Dungean semantics and critical questions are modeled as exceptions. The deliberation problem was represented as the statement *there exists an equitable solution to the case*. For resolving the equity issue, three alternatives are modeled:

- Giving the ball to **Popov**;
- Giving the ball to **Hayashi**;
- Selling the ball and **dividing** the proceeds **equally** between Popov and Hayashi.

While the first two alternatives are rejected with rather formal means, the third one seems rather obvious and intuitive, and therefore tricky to model in a legal setting. The authors rely on the principle of equitable division, sustained by three arguments:

– An **argument from authority**, citing a law review article by professor Helmolz which proposed applying the principal of equitable division to resolve disputes between finders of lost property;

– An **argument from legal tradition**, pointing out that the principal of quitable division has its roots in ancient Roman law;

– By citing **two precedent cases** in which the principle of equitable division had been applied.

The authors also note that, in supporting these arguments (three of which are from practical reasoning, and one from tradition), premises had to be left implicit because of lack of space in the diagram. For example, stating that dividing property equally among persons with equally good claims to the property promotes the value of equity, is missing. But as evidenced by the legal argument about the authority of a judge to divide property in such cases, clearly not all

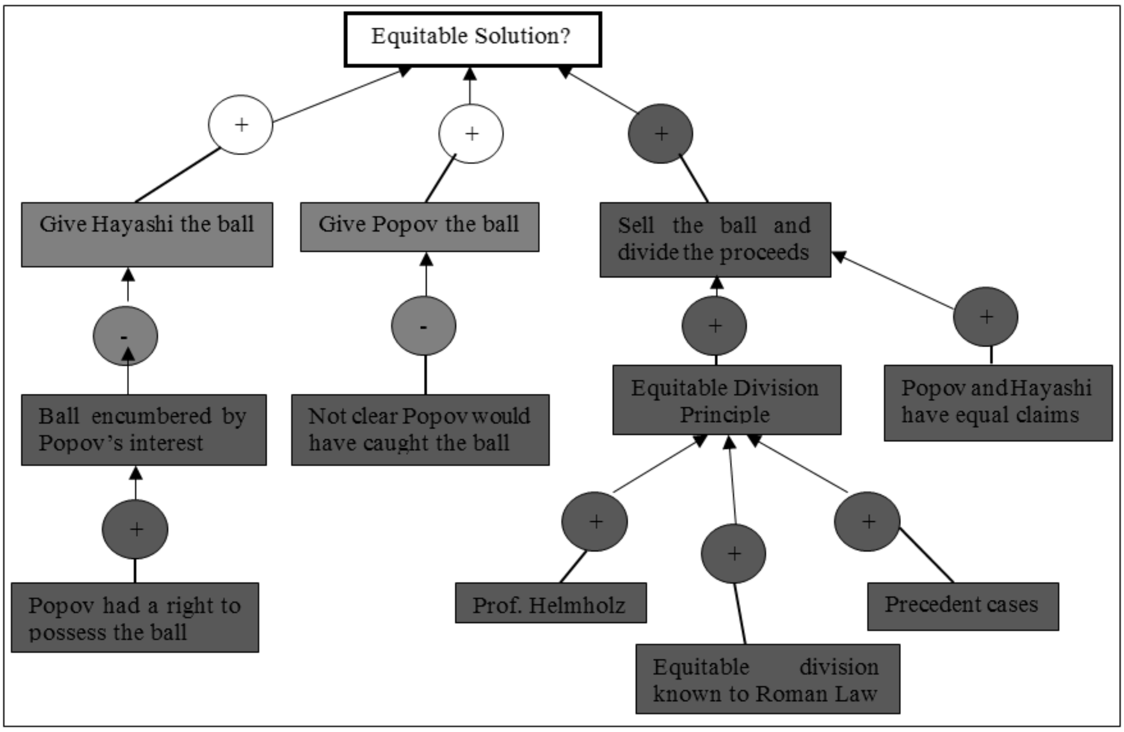


Fig. 5.12 – Graph representing the final solution in [Gordon and Walton 2012] (light grey equals to refused, dark grey equals to accepted).

arguments in a deliberation need be arguments from practical reasoning.

5.6. Requirements for a Rule Markup Language

5.6.1. Introduction

Following is a review of all the elements which are fundamental in a proper knowledge representation framework for the legal field. These elements have all been presented in chapters 3, 4, 5, and therefore they will not be fully explained, but simply recalled in order to list them and to provide easy reference for the evaluation of present (and future) efforts in this field.

5.6.2. Ontological requirements

5.6.2.1. Four-Layers classification of legal documents

When constructing a map of concepts, it is fundamental to be able to disambiguate between *pure*, abstract concepts and mere instantiations of the concept, or *tokens*. This should be done, however, without taking a specific perspective: it may well happen, in fact, that the concept to be modeled is abstract in one case, and concrete in the next case. A layered model for legal documents should therefore include IDs for at least the first three FRBR layers: Work, Expression, Manifestation. The Item category should be used in the referral to mere documents (not legal acts) when the focus is on the specificity of a single instance of that document (i.e. clauses appearing only in the consumer's copy of the contract).

5.6.2.2. Connection to Linked Data

The commitment to the community, and the mix of advantages and duties it entails, is not a mere modelling choice, nor a strategy or an approach: it is, evidently, a philosophical shift in the concept of scientific production of our age.

It would be nice to have all legal documents modeled by the same, single legal expert, possibly a world-renown leader of the legal doctrine in that legal field. Unfortunately, this is not feasible since the cost of such an effort would be excessive. Once upon a time, at the beginning of the last era, knowledge was collected by noble people who had not a *job* in the sense that this word has nowadays. Encyclopedias were built this way. In our times, we reached a far opposite goal: that of perfectly dividing labor between a large number of people. Still, however, such works are undertaken by single people. These are monographies which do not pretend to be up-to-date. In the light of these considerations, we realize the need to rely on the network and on the chances brought forward by social platforms, which allow everyone who's willing to join the effort to do so, while at the same time keeping an eye for quality through a peer evaluation system. A such built system would, in theory, represent a better tool than legal doctrine: it could, in fact, take the place of law reviews in enriching the laws with comments on their (common and uncommon) interpretations and applications. This is the reason why the legal ontology described in chapter 2 should become part of the Liked Data initiative. It would be a way to manifest the intended approach of this kind of legal data collection.

At the same time, the legal ontology would get benefit from several databases which are already part of the Liked Data. In particular, the databases of Liked Data from which the ontology would best benefit would be official law collections such as data.gov.uk in the United Kingdom. In Italy, the official online repository of www.normattiva.it is an excellent source of structured data. As noted by [Tiscornia et al. 2011], the adoption of the Linked Data principles for publication of legal data, joined with existing efforts of standardization in the identification and representation of legal information, would open the way to a whole range of innovative legal services and applications based on top of a *Legal Data Cloud*.

In order to exploit the Linked Data network, the ontology would need dereferenceable URIs, namely URLs. Emerging standards for legal rules representation such as LegalRuleML should ensure legally valid references through full support for W3C's and IETF's URIs and IRIs as prefix in front of the community's standard names.

5.6.2.3. Isomorphism

In knowledge representation, it is necessary to clearly establish the relationships intercurring between the form and the substance. In the legal field, this is much more important, since in that field the form drastically influences the substance. For a proper representation, isomorphism should thus be ensured in both directions: from the data to the text fragment and from the text fragment to the data. It must be possible to link multiple legal concepts to the same text fragment, and at the same time to indicate the same text fragment as expressing several, distinct legal concepts. See 5.6.1. for how LegalRuleML achieves this, and how this solution can be pointed at as an example of good practice for this aspect.

5.6.2.4. Logical representation of advanced legal concepts

Advanced legal concepts derive from complex social processes which lasted millennia. A full account for the semantics of these concepts cannot be fully provided by a single (or a small set of) interpretation(s), formally coded responding to general criteria. In other words, law is so full of exceptions, distinctions, stratifications, that every categorization involves a certain loss of semantics. Therefore in an accurate representation very legal concept (legal figure, or legal category) should *speak for himself*, being individually described by a set of logic formulas (and, possibly, even some specific logical modules in the modal logic environment). This means not only that rules have to be specifically modeled upon the norms they are due to express, but also that the categorization axioms (the ontology properties) have to give account for this specificity.

In a linked, social perspective, the modelling choices can be saved as *ontology design patterns* [Gangemi 2007] in order to be discussed, evaluated, classified and reused. This approach would bring two substantial benefits:

- Building a collection of legal design patterns that can be polished, improved and updated over time to reach a consensus on the possible meanings of advanced legal concepts;

- Providing a set of described, evaluated solutions to the legal expert facing modelling issues.

The example given for efficacy suspension in 5.3.3. is meaningful in this perspective: distinguishing the suspension frame into efficacy inclusion and efficacy exclusion forms is a modelling choice, requiring a certain extent of legal awareness. These choices, despite being rather general, are not an immutable truth on the law but simply the representation of a common procedure in Italian legal drafting. Modelling these peculiarity is necessary to achieve a complete interaction between the text and the norm, and modelling them in a customizable, authored way is necessary to provide full control over the process, removing the constraints to the modelling choices and thus setting the necessary site where conflicting premises on legal interpretation (in other words, the legal doctrine) can be stored.

5.6.2.5. The Middle-Out Approach

Observing legal knowledge from the perspective of ITs, it turns out that its construction is in fact a mix of the approaches called *bottom-up* (case-law is a paramount of such a methodology) and *top-down* (legal doctrine, legal philosophy, and mechanisms such as law-decrees or laws enacting EU directives). These two approaches cannot be clearly distinguished in all cases, and it happens that the same legal document presents a contribution to the legal knowledge which is itself a mix of these (i.e. a decision of a constitutional court typically combines arguments from legal principles and arguments from precedents).

In the semantic framework, there is no space for distributing these modelling perspectives to different layers of the Semantic Web stack. It is always in the ontology layer that the two approaches need to be mixed. In chapter 3, the two approaches create a distinction between the core and domain ontology modules, where the *core module* holds general concepts and elements functional to the shallow reasoning, and the *domain module* holds the collection of the precedents and an extensive list of all their elements, from the material circumstances put to the attention of the judge to the legal statuses and norms they evoke.

It is, however, just a functional distinction which doesn't mean much: the core and domain concepts are closely related, and there are very few concepts which cannot be altered in a legal knowledge representation. The core module, therefore, cannot include all the legal concepts derived from a top-down approach, because some of them represent a perspective and therefore have to be included in subsets of the legal ontology to prevent inconsistencies with other perspectives on the same concepts. Shrinking the range of the core, immutable module of the ontology, it becomes clear that it is the domain ontology module which carries the burden of representing the legal perspectives. These modules will therefore contain a substantial part of the top-down knowledge base, as well as the whole bottom-up content.

5.6.2.6. Description Logics

Choosing the logics for the representation of legal knowledge is an extremely complex task. The implication of the various logics (and of theirs subsets) are wide and sometimes difficult to recognize, requiring advanced study of the subject or a meticulous test of the various solutions. Fortunately, the Semantic Web community already reached a certain degree of understanding (and shared standards) at the Ontology Layer, and we can then refer to the choices of the W3C on the matter. In particular, we can rely on their choice of adopting OWL-DL Description Logics for their OWL language (even if the notation of OWL is more close to first-order logic, the logics itself is undoubtedly a subset of description logics). In particular, the language of OWL2 is DL $\mathcal{SROIQ}(\mathcal{D})$. These letters stand for a set of characteristics which define a subset of the description logics language. Following is a complete list of the features of this language (please note that the letter $\mathcal{S}$ includes both $\mathcal{AL}$ and $\mathcal{C}$).

- $\mathcal{AL}$ = attributive language, the base language, which allows:
  - atomic negation (negation of concept names that do not appear on the left hand side of axioms);
  - concept intersection;
  - universal restrictions;
  - limited existential quantification;

- $\mathcal{C}$ = complex concept negation;
- $\mathcal{S}$ = transitive roles;
- $\mathcal{R}$ = limited complex role inclusion axioms; reflexivity and irreflexivity; role disjointness;
- $\mathcal{O}$ = nominals. (Enumerated classes of object value restrictions - owl:oneOf, owl:hasValue);
- $\mathcal{I}$ = inverse properties;
- $\mathcal{Q}$ = qualified cardinality restrictions (cardinality restrictions that have fillers other than `owl:Thing`);
- $(\mathcal{D})$ = use of datatype properties, data values or data types.

Any knowledge representation language pretending to build on the achieved standards in the community must come to terms with these characteristics. The present research, in fact, already exploits most of them:

- **Atomic negation** is fundamental to avoid the instantiation of negative entities (i.e. “NotSpecificallySigned”), but atomic negation hinders reasoning due to the open world assumption;
- **Concept intersection** is widely implemented in class intersections to define complex classes;
- **Universal restrictions**, as distincted to existential restrictions, are also widely used in legal concepts definition (i.e. a distinct signed box containing *only* oppressive clauses implies a specific signing);
- **Limited existential quantification** allows a generic quantification of elements in the ontology, and is therefore a basic building block of the ontology;
- **Complex concept negation,** the negation of classes as part of an axiom, is widely used in legal relevancy, i.e. (`contained_in (not)SpecificallySigned`);
- **Transitivity** is important in the actual ontology design, which relies a lot on properties to link material circumstances to legal statuses. The same can be said about **limited complex role inclusion axioms** and **inverse properties**;
- **Qualifyed cardinality restrictions** are not yet exploited in the ontology, but they represent a cornerstone for modelling themporal parameters (i.e. the period of validity of a certain representation of a norm);
- **Datatype properties, data values or data types** is obviously useful in quantifying elements such as sums of money and time parameters.

5.6.2.7. Reasoning

The legal ontology should be capable of performing shallow reasoning on the *static* elements of its knowledge base. Such reasoning functions (in particular, reasoning on legal relevance) should constitute a set of modules to the ontology, to be used only if necessary: the risk of creating conflicts with defeasible rules seems too big to pretend reasoning axioms to be the skeleton of the ontology. The range (and purpose) of these modules are two:

Data completion, to automatically fill the net of data related to a law, a legal case, a judicial sentence. In order to ensure a full isomorphism with the legal text, in fact, the information asserted in the ontology is only that which can be directly linked to the source document. The problem is that many legal documents cite other pieces of information without providing a full range of informations about it. DL helps to overcome this problem, automatically assigning, on the basis of the type of document, "default" values to fill the missing data. In some cases (when the document has a simple, immutable content), this simple data completion activity can be "embedded" directly into the base ontology. In other cases (when we are completing data about a document which evolves over time), this function may cause redundant or inconsistent data.

Reasoning on legal relevance allows to perform advanced queries on the ontology. The complexity and depth of the reasoning is not comparable with that resulting from the application of defeasible rules, but the OWL reasoning has the advantage of taking place within the KB itself. In other words, some information about the content of the legal document (the outcome of the judgement, the kind of interpretations performed, the concepts of law involved) could be calculated *on the fly* while performing searches on the database, without involving defeasible reasoner in that phase. Later, the results coming from this reasoning can be explored using argumentation to perform complex tasks. For a framework due to be used in the Semantic Web, such a differentiation of tasks between layers of the semantic stack could prove precious for an optimization of resources and an improvement in computability.

5.6.2.8. Modularity

For a tool managing interpretations and perspectives on legal phenomena such as the present project, to include all the contributions at once is not the optimal solution. This is not due only to computability issues (complex axioms for reasoning on legal relevance can include more than five variables: reasoners cannot compute all the possible combinations of a large set of statements in reasonable time): in fact, the different contributions – even on the same legal subject or issue – may lead to undesired or inconsistent implications. It is therefore necessary that the single contributions get *wrapped* in several internally consistent ontology modules, which refer to the core ontology but are otherwise autonomous from each other. In this way, the tools managing the knowledge base can import one or more of these modules and merge them into a single ontology before running the reasoner. In case of inconsistencies or inefficiencies in reasoning on the merged ontology, the tool can take a step back and conduct autonomous reasoning on the single ontologies, highlighting the inconsistencies. It is then possible to *switch off* some of the reasoning features of the ontology set in order to provide a consistent knowledge base to the rule layer. It is also possible to conduct argumentation on an inconsistent knowledge base, without removing the features causing the inconsistency, but in that case the defeasible reasoner should conduct autonomous reasoning (and thus build autonomous graphs) on the single ontology modules. These modules can then be presented together to the user as a single graph, highlighting the overlapping and the inconsistent parts.

To achieve this modularity, an important help may come from the reasoning capabilities of OWL (on the contrary, it is not possible to rely on defeasible logics since the construction of the knowledge base has to take place within the ontology layer of the Semantic Web cake, thus involving only the standards achieved in that sector). The ontology reasoner can in fact detect the concepts which are present in each module, and therefore extend the knowledge base by merging other modules into the initial selection, on the basis of their contribution to the issue which is being analyzed at the moment.

5.6.3. Syntactical requirements

5.6.3.1. Defeasible Logics

If compared to the ontology layer, the rule layer is far from constrained by consistency. This is well received, as every representation of legal reasoning must be capable of managing explicitly inconsistent information, since (at least apparent) inconsistency is a constituting part of legal doctrine.

When building the knowledge base with OWL ontologies, it turns useful to introduce disjunctions between classes and negations, in order to enhance the reasoning capabilities of the ontology sets. These disjunctions, however, revert to disadvantages when conflicting legal rules (laws, agreements, or judicial interpretations) form the basis for rule-based reasoning. It can well happen that a contract is labeled as “signed” and “not signed” contemporarily, as a result of the application of two different rules. In order to perform effective reasoning on the case, however, it is necessary to (have the opportunity to) decide which rule to apply to the specific case: this is allowed by distinguishing rules between strict rules, defeasible rules and defeaters.

Defeasible logics allow not only to resolve conflicts between rules whose heads are the opposite one to another, but also to search the rule base for specific kind of rules: for example, the user can look for rules which explicitly defeat an undesired legal consequence, or to distinguish between rules whose activation will imply some legal consequence despite any conflicting norm from those whose provision will be valid only unless other rules apply.

5.6.3.2. Deontic Logics

It would be impossible to model the content of legal provisions without deontic operators, as they add necessary semantics to basic actions, providing not only modalities of, but also reasons for them. [Governatori and Rotolo 2008] proposed an extension of Defeasible Logics to capture combinations of mental attributes and deontic concepts, as well as consequence relations. To add deontic logics means to extend defeasible logics with some particular modal operator. In doing this there are two options:

- To use the same inferential mechanism as basic defeasible logics and to represent explicitly the modal operators in the conclusion of rules;
- To introduce new types of rules for the modal operators to differentiate between modal and factual rules.

The differences between the two approaches are three:

- In the first approache there is only one type of rule, while the second accounts for factual and modal rules;
- In the first approach needs the introduction of a set of *p*-incompatible literals for every literal *p*;
- In some cases, the first approach must account for *conversions* – rules that permit to use a rule for a certain modality as it were for another modality – which make it less conceptual than the second approach.
- The second approach can use different proof conditions based on the modal rules to offer a more fine grained control over modal operators.

For these reasons, it seems the case to adopt the second approach, called Modal Defeasible Logic and already adopted in [Lam and Governatori 2009] for the SPINdle rule engine.

5.6.3.3. Dungean Semantics

The application of defeasible logics brings a reduction of the implications brought forward by a set of rules, which do not apply indistinctively to the whole knowledge base but rather interact with each other by destroying bonds which other rules would otherwise create. This reduction of information is functional to a subsequent phase of the argument construction from rules: the phase of the argument evaluation.

When all the rules which compose the rule set have been applied, a set of valid arguments is presented, each of them stating (or negating) one or more statements (the consequence) if one or more statements (the premises) meets the rule's requirements. These rules interact with each other (for example by providing valid premises for other rules to apply), and a net of statements linked by the rules is formed.

For a series of reasons, however, it is possible that several argument do not "trigger" (they do not appear as valid) even if the major, explicit premises are true. The cause may be that a minor premise is not explicitly stated as true (lack of metadata, enthymemes), that some rule which would defeat the present argument is still *applicable* (means that its requirements are not yet met, but neither excluded), or that the fact to be proved is so basic that its truth cannot be derived by means of other rules (*regressio ad infinitum*).
In those cases, a strict application of the negation-as-failure principle would drastically reduce the effectivity of the argumentation.

For this reason, it is fundamental that the argumentation layer follows Dungean semantics, in order to maximize the number of statements triggered as a consequence of the application of the rules to the knowledge base. This prevents that some interesting results remain buried under the surface of the arguments graph (avoid false negatives), while it is rather simple to polish the results by excluding the arguments which appear into the argumentation even if none of its premises is directly concerned in the specific case (false positives). The same approach could be applied to the choice between ambiguity blocking and ambiguity propagation (see 6.2.1.1.)

5.6.3.4. Argumentation Schemes

Support for argumentation schemes is necessary in order to correctly instantiate rules into arguments, highlighting the implicit premises through critical questions. The solution can be used to merge the different contributions to the argument graph, correctly nesting the conclusion into a net of premises and exceptions. Oversimplified interpretation rules will then reveal their weaknesses, if the knowledge base fails at providing information to answer the critical questions.

Argumentation schemes can also be used to model the process itself: instead of managing the contribution to some abstract argument (i.e. towards the validity or invalidity of a contract with certain characteristics), they can be used to complete the scheme which represents the reasoning of the judge in a specific case. This is also useful to reveal missing parts in the decision's groundings.

It is also worth exploring the possibility of extending the concept (and use) of argument schemes beyond the mere classification of [Gordon and Walton 2009]: argumentation schemes could, in fact, be used to model the types of intervention that the parties (or the judge) can operate into a process. For example, a competency regulation claim could evoke a specific argument scheme which reveals implicit premises about when and how such a claim can be raised in a process. This use of argumentation schemes would allow a complete management of the trial as a dialogical process, and enhance the interaction of semantics and syntax for those specific claims (which are labeled in the ontology and consequently elaborated through rules and argumentation schemes). It has, however, the disadvantage of exponentially increasing the number of schemes, which contradicts the spirit behind the concept of argumentation schemes (intended to be a relatively small numbers of patterns of reasoning, and not an exhaustive list of legal dialogical tools).

5.6.3.5. Proof Standards

Proof standards naturally complete the argument scheme approach, providing a complete representation of argumentation processes. When managing a single legal dispute, proof standards allow adding information on the single arguments (claims and counter-claims), assigning weights to arguments and proof standards to premises. Balancing the weights of the single argument until the result of their combination coincides with that of the real case, the user makes explicit information which was before implicit (sometimes even to the judge which weighted those arguments). This kind of deduction is an important elicitation that turns very useful when trying to use those arguments outside their original case: it is in fact possible to create a "mixed" case when different arguments from different precedents are put forward to prove a single claim, common to all the precedents. This brings to interesting results in case-law comparison, and can be used to represent factors and dimensions.

Similarly to argumentation schemes, also proof standards can be extended to represent the authority of the contributions: case-law, jurisprudence and law can be classified on the basis of their strength (there is stronger case-law, stronger jurisprudence, stronger law) and this strength can be represented through different proof standards assigned to arguments which defeat those contributions. In this way, different positions on the same legal issue can be weighted and the tool be exploited as a law evaluation environment.

5.6.3.6. Factors

In the framework built during the present research, the class *legal status* was modeled upon the concept of factors as described in [Bench-Capon 2012]. In theory, it is possible to use the legal status class to model factors of cases, and to use them to reason on case-law. However, factors are intended as a further level of granularity than legal statuses. Providing information about every factor for every element of the ontology is not viable (due to knowledge acquisition bottleneck and computability issues – not to mention the lack of relevance of a large part of them). For this reason, there should be no strong constraint depending on factors on the ontology set: in this way, there is no risk that meaningful inferences are prevented from lack of metadata concerning factors. On the rule level, however, it is possible to rely on factors to evoke particular arguments. In case of lack of information, the reasoner could nevertheless suggest factors that, if present, would constitute the basis for factor-based arguments.

5.6.3.7. Dimensions

Dimensions behave differently than factors or values. They do not simply distinguish from presence or absence of an element, but rather set a graduated scale between two extremes. The concept was introduced for analysis of case-law, but it can be used whenever a comparison has to be made between elements which do not have a fixed value assigned, but whose relative placement is the only relevant information available.

This distinction is similar to the design dilemma on argument weight: is it better to assign an arbitrary value to each argument, using these values to perform a (partial) ranking, or should we explicitly place each argument in an ordered list, from the strongest to the weakest? The issue was analyzed in 4.5.9., and here it is just the case to recall how both solutions should be implemented in the system, and the reasoner engine should be able to mediate between the two, i.e. compiling a ranking on the basis of the arbitrary values (and resolving the tie-related issues), or – on the contrary – assigning increasing values starting from the last argument in the ranking.

5.6.3.8. Values

Values can be treated like factors: their information can be exploited as long as it is present. Therefore, no constraints on the ontology set should involve values. On the rule layer, it is possible to build arguments which rely on values, and they can be suggested by the reasoner if no information on the values is present.

For an efficient representation, it would be useful to evoke, together with the argument whose value is present (or missing), also counterarguments and undercutters. In this way, the user can properly evaluate the possibility that others values are missing in the ontology representation, values which would prevent the validity of suggested argument. It is important to distinguish between the different approaches to argument analysis (factors, dimensions, values). These data can be used together, but not carefree if we want to keep a straight approach to the analysis of legal phenomena, and avoid the mix of heterogeneous arguments and theories.

5.6.3.9. Temporal reasoning

Temporal parameters are fundamental for an automatic management of law sources, and for a correct representation of the applicable norms at the time of the judicial decision. Temporal reasoning allows to highlight the part(s) of the decision which have been overruled by more recent laws or decisions. As exhaustively described in 5.3.2., temporal parameters should be given proper attention in the metadata, ontology and rule layers.

5.6.3.10. Procedural argumentation

The analysis of judicial decisions cannot assume a synchronic perspective, or it will be unable to explain the invalidity of certain arguments brought forward during the trial. Judicial decisions give account for what happened during the hearings at the court, hearings which are in turn regulated by legal procedure. In order to represent those dynamics, it is necessary to assume a diachronic, procedural perspective. The argumentation graph should therefore be multi-staged, representing different phases (steps) of the legal process and using information such as the absolute and relative stage in the evaluation of arguments. For example, an exception which can be proposed only within a certain stage of the process could be evaluated as invalid on the basis of the absolute stage of the dialogue; a counter-claim could be evaluated as invalid on the basis of the relative stage of the dialogue (i.e. if the claim to be countered was introduced at a much earlier stage of the process).

The metadata for such stages can be modeled as a subset of temporal parameters. Instead of using absolute values, the stages are defined (t1, t2, …, t*n*) and then recalled by the instances of the KB representing the various events of the trial. The reasoner will use these parameters to apply procedural rules, and the outcome will be presented as a multi-stage argument graph.

The argumentation graph should not only give account for the stages of the process, but also for the tasks of the parties. This means that argument graphs may not be directed towards a single conclusion. Depending on the stage of the process and on the party performing the argumentation (plaintiff, defendant, civil party, judge…) every contribution may have a distinct task: not only that of proving the main claim or undermining the opponent’s argument (either directly, or indirectly through an attack on its premises), but also that of asking an external assessment (a probe, a preemptive appeal to the constitutional court…) or of modifying the *quantum* rather than the *an* (which means, not trying to prove that no refund is needed, but that the sum to be refunded is different). Representing such diverse tasks (even if not in a single graph) may overcomplicate the representation, nevertheless it is fundamental in order to catch the different purposes of the various contributions (for example, a doctrinal contribution may simply introduce a consideration to some argument or solution, without influencing the acceptability of that argument in any way whatsoever).

5.6.4. A table summarizing the requirements for a rule markup language

Following is a table that summarizes the requirements described so far, together with an indication of how the present project (based on OWL and on the Carneades Argumentation System) meets/fails to meet them.

Ontological requirements	**Judicial Framework “*Judging Contracts 1.0*”** **Carneades Argumentation System**

Four-Layers classification of legal documents	The LKIF ontology does not support such distinction. The system could anyway be extended in that direction.
Connection to Linked Data	The new Carneades Policy Modeling tool is based on a web-based interface, therefore providing an approach which favors knowledge interconnection.
Isomorphism	LKIF ensures a good isomorphism with its devoted classes such as medium and expression.
Logical representation of advanced legal concepts	Nor LKIF or Carneades provide support for NLP tools. Semantic construct to represent advanced legal concepts and their language, however, are possible.
The Middle-Out Approach	Carneades allows the import of knowledge from case-based reasoning as well as from manually inserted knowledge.
Description Logics	Carneades successfully imports knowledge from OWL ontologies, but (at the current state) does not distinguish between stated and inferred knowledge.
Reasoning	Carneades is able to import reasoning performed by the ontology. It is also possible to translate OWL's axioms into LKIF-Rules.
Modularity	Carneades allows importing several files (ontologies and/or rulebases). This permits to build the KB by combining elementar modules, and to reason on all of them together.
Syntactical requirements	
Defeasible Logics	Carneades supports defeasible logics, but without a priority relation. It relies, instead, on weights.
Dungean Semantics	Carneades represents Dungean semantics, or, at least, it can be easily replicated in the argument visualizer tool.

Argumentation Schemes	Carneades supports argumentation schemes, even if automatic weight assignment did not work in version 1.0.2.
Proof Standards	Carneades supports assigning proof standards to arguments, even if this did not ensure automatic inferencing in 1.0.2, due to the lack of automatic weight assignment.
Factors	Factors can be represented in the Carneades system.
Dimensions	In order to perform a reasoning on dimensions in judicial cases similar to [Bench-Capon 2012] the reasoner, given the rules, the facts and the conclusions of reasoning, should produce the (best possible) priority relation between the rules. This seems difficult to represent in Carneades, which has no priority relations between the rules but rather weight assignment.
Values	Values could be represented similarly to factors
Temporal reasoning	Temporal reasoning is limited in the Carneades System.
Procedural argumentation	Procedural argumentation is represented in the Carneades tool, and a sequence of reasoning procedures with different goals is possible inside the application.

5.6.5. The Reasoner

Considering the importance of combining two different approaches and two layers of the Semantic Web stack, the pivot of the whole framework is the reasoner, combining the ontology with the rule set to reproduce the argumentation process. In the light of the issues related to the use of Carneades, aspresented in 3.5., it is necessary to design a hybrid reasoner which can handle different kinds of reasonings and combine them to compile a multi-staged argumentation graph.

The next chapter concludes the research by analyzing the requirements of such a hybrid reasoner and the existing projects in that field, highlighting strengths and weaknesses.

5.7. LegalRuleML

[Palmirani et al. 2011c] presents LegalRuleML, an extension of RuleML[27]. LegalRuleML enhances legal rules modelling by introducing fundamental features such as isomorphism, defeasible logics, jurisdiction and authority, legal temporal parameters, legal deontics operators, qualifications, semantic of negation, behaviors. The language also allows the inclusion of elements and statements compliant with external ontologies.

The syntax and structure of this language are a work in progress: therefore all tags, elements, attributes recalled here follow syntax proposals presented in the TC, and some of them may not correspond to the definitive syntax of LegalRuleML[28]. The requirements expressed are nevertheless important for tackling some of the problems identified in the present chapter. The main characteristics which will be examined in the present research are isomorphism and defeasible logics, but also the features of qualification, behaviors and semantic of negation will also be addressed - as they already were in Chapter 4. Deontics represent a necessary addition to the rule modelling, in order to better represent some concepts while at the same time maintaining isomorphism, and they will be implemented successively. A proposal for implementing argumentation schemes is then presented. Finally, temporal parameters are proposed, as already presented in 5.3.2.: the concept of temporal dimensions of norms is crucial to legal knowledge representation, nevertheless it represent a complex system and parallel to the present research, concerning norm applicability on a subsumptive rather than on a temporal parameter.

5.7.1. Achieving isomorphism

The `<ruleInfo>` section of the LegalRuleML standard encloses all information needed to effectively bind the rules to the text. First, the `<identifications>` and `<references>` elements give ad ID and a URI to the authors of rules and to the source documents respectively. The `<sources>` element recalls these IDs both for entire rules (referring to the ID of the parent element `<implies>`) and even for the single statements rules are composed of (referring to the ID of the children elements `<atom>`). Following is an example of `<ruleInfo>` section:

```
<lrml:ruleInfo id="ruleInfo2" appliesTo="#rule2">

      <lrml:sources id="sourceBlock2">

            <lrml:source element="#atom1" idRef="#art1341-com2"/>

            <lrml:source element="#atom2" idRef="#art1341-com1"/>
```

[27] RuleML is an XML based language for the representation of legal rules using formal semantics. See [Lee and Sohn 2003].

[28] Documents of the OASIS LegalRuleML TC are available at https://www.oasis-open.org/committees/documents.php?wg_abbrev=legalruleml. The mailing list describing the work in progress can be browsed at https://lists.oasis-open.org/archives/legalruleml/.

```
            <lrml:source element="#atom3" idRef="#art1341-com1"/>

            <lrml:source element="#atom4" idRef="#art1341com2"/>

        </lrml:sources>

        <lrml:strength iri="&dfsont;defeater"/>

        <lrml:jurisdiction iri="&jurisdictions;italy"/>

        <lrml:author idRef="#aut1"/>

        <lrml:times idRef="#t1"/>

        <lrml:creationDateTime idRef="#e1"/>

    </lrml:ruleInfo>
```
[29]

Inside the `<ruleInfo>` section, which defines a set of variables to be recalled by single rules (i.e.: `<implies id=”rule2” refersTo= "ruleInfo1">`), two children elements are particularly fit for enhancing isomorphism between the rule system and case-law:

- `<jurisdiction>` semantically specifies the national (or regional) scope of the norm, linking to an element in the ontology. For the purposes of the present research, it is very useful in distinguishing between local, regional, national, european and international law, and also in determining the scope of precedents distinguishing between the different regional courts.
- `<authority>` semantically specifies in the same way which legal authority gave force to the legal text. The efficacy and applicability of the legal rule is a direct consequence of this enacting power (represented in the LKIF-Core ontology by the "attitude" class). This element is fundamental to model all the information necessary to distinguish between different precedents, i.e. giving priority to Cassation Court and Constitutional Court decisions.

Modelling the legal rules in the LegalRuleML language highlighted the potentialities of this tool in achieving isomorphism and representing defeasibility and temporal parameters. In particular, the `<ruleInfo>` section introduces detailed information on the context of the rule. This rule-centric metadata approach favours the isomorphism with the legal text when facing changes in the source documents, and the localization of all information related to the rule in a unique XML node. In this way it is possible to explicitly refer to the source documents of each part of a rule, also when a rule takes origin from multiple documental sources, allowing the engine to understand which part of the rule comes from which document. At the same time, different rule authors are allowed to model the same text fragment in different ways, being always clear which author modelled which rules on a certain legal document. As it should be clear, this system represents the ideal background for a social tool reproducing legal doctrine, where different jurists model their version of the norm by writing slightly different rules (in terms of legal

---

[29] The proposal can be found at http://www.oasis-open.org/apps/org/workgroup/legalruleml/download.php/45887/2.8isomorphism.001.doc.

concepts, deontics, or procedural dynamics) for the same (combination of) legal text fragment(s).

5.7.2. Introducing elements of defeasible logics

LegalRuleML includes, in the `<lrml:Context>` section, the `element <lrml:appliesStrength iri="#type"/>` which defines the type of each rule in the defeasible logics dynamics (types are strict/defeasible/defeater). Priority relations are built through `the <lrml:Overrides over="#rule2" under="#rule2"/>` element[30], in the following form:

```
<lrml:overrides id=”ovr1”>

      <lrml:rule keyref="#rule_3"/>

      <lrml:rule keyref="#rule_2"/>

</lrml:overrides>
```
[31]

This implementation of defeasibility should allow a better management of exceptions than in LKIF-Rule, where exceptions had to be made explicit in the rule syntax. If an exception presents itself in the form of two legal fragments not explicitly referring to each other but rather disposing opposing legal consequences (i.e. efficacy vs. inefficacy) it is possible to model these rules independently, and then create a priority relation reflecting the actual hierarchy between the two norms. Moreover, this solution allows a relative management of hierarchy, without the need to assign arbitrary "weight" values to each rule. However, this approach has some backsides: see 4.5.9. for details.

5.7.3. Introducing temporal parameters to the Rule

The `<lrml:Context>` section permits also to assign `<lrml:appliesTemporalCharacteristics keyref="#tblock1"/>` element, not indeed a normal attribute but rather a section introducing a whole different layer: it contains information on the time periods of the legal rule’s coming into force, efficacy, application. The representation of *temporal dimensions* of legal rules using three axes is a crucial addition towards the automatic management of legal rules in connection with their legally binding

[30] This tag is developed jointly with RuleML (http://ruleml.org/) and in particular with the Defeasibility RuleML TG (http://ruleml.org/1.0/defeasible.html).

[31] This implementation of defeasibility should allow a better management of exceptions than in LKIF-Rule, where exceptions had to be made explicit in the rule syntax. If an exception presents itself in the form of two legal fragments not explicitly referring to each other but rather disposing opposing legal consequences (i.e. efficacy vs. inefficacy) it is possible to model these rules independently, and then create a priority relation reflecting the actual hierarchy between the two norms. Moreover, this solution allows a relative management of hierarchy, without the need to assign arbitrary "weight" values to each rule. The proposal can be found at https://www.oasis-open.org/committees/download.php/46454/2.1.1defeasibility.0 06.doc. Meaningful comments by TC member Tara Athan can be found at http://www.oasis-open.org/apps/org/workgroup/legalruleml/download.php/45888/2.1defeasibility.002.002.doc.

documents [Palmirani et al. 2010], and LegalRuleML allows to specify these time coordinates for each rule, starting from the identification of the relevant points in time through a list of `<lrml:TimeInstant>` such as:

```
<lrml:TimeInstant id="e2" value="1942-04-21T01:01:00.0Z"/>.
```

LegalRuleML allows to specify these time coordinates for each rule. Events' IDS are recalled by the `<timeBlock>` element, which adds information on the event which occurs (`start`, `end`) and on the axis which is affected:

```
<lrml:timeBlock id="t1">
     <lrml:time start="#e2" refType="&lkif;#efficacy"/>
</lrml:timeBlock>
```
[32]

In this way it is possible to reach a deep granularity of annotation not possible in LKIF in order to enrich each parts of the rule (body, head, atom) with the temporal parameters.

5.7.4. Semantic qualification of Negation

Distinguishing the concept of *negation as failure* from that of *explicit negation* is important in the legal field. Moreover, the representation of legal knowledge made by the present research has to deal with both the semantics when filling the gap between the text representation and the normative syntax. LegalruleML contains the modules neg_module.xsd and naf_module.xsd which can be linked to express de desired semantics (or a specific concept ontologies through the attribute `refersTo`).

```
<xs:attributeGroup name="Neg.attlist">
     <xs:attributeGroup ref="refersTo"/>
</xs:attributeGroup>
<xs:attributeGroup name="Naf.attlist">
     <xs:attributeGroup ref="refersTo"/>
</xs:attributeGroup>
```

5.7.5. Deontics

The LegalRuleML TC is now working on the deontic operators and behaviors using specific elements like `<lrml:Obbligation>`. Also LKIF permitted deontic operators, but they were not connected with temporal parameters and textual origin sources that was a limit in the

[32] The proposal can be found at http://markmail.org/message/oy34tkzzr3r2ldhz?q=temporal+lis t:org%2Eoasis-open+list:org%2Exml+list:org%2Eebxml.

Carneades proof-of-concept. The operators needed to manage deontic logic and behavious like *violation* and *reparation* are defined in the *legal_operators* module, shown in Figure 5.13.

Behaviors represent a particular sequence of deontic operators that starts with an *obligation* or a *prohibition* and ends with a *permission*. The *violation* is a unary relationship that refers to the *obligation*/*prohibition* subject of the violation. The *reparation* is a unary relationship providing a link to the relevant penalty. These particular behavioral patterns are not modeled in LKIF. In this perspective, normative statements are also divided in two main categories following the theory of law: *prescriptive statements* `<lrml:PrescriptiveStatement key="ps1">` and *constitutive statements* `<lrml:ConstitutiveStatement key="ps1">`.

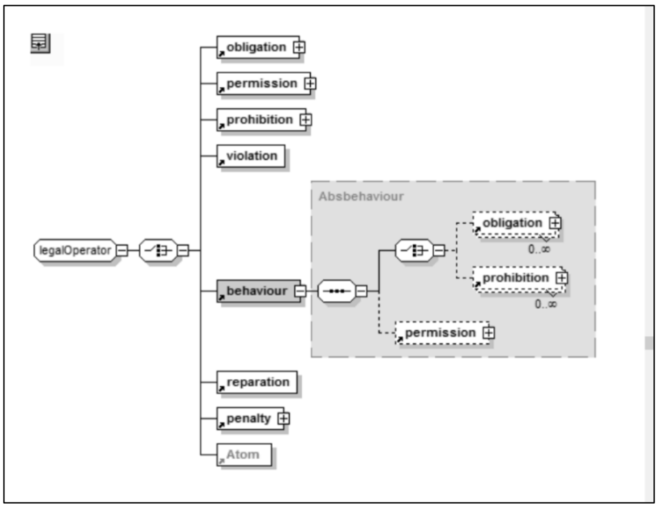


Fig. 5.13 – Model of deontic logics in LegalRuleML

Following is an application of the system to the present framework:

To model the content of a contract, the first step is to model the rule which contains in the conclusion a prohibition to sell the same product to third parties:

```
<Implies id="Exclusive">
    <then>
        <prohibition>
            <Atom id="Excl-prh1-atm1">
                <Rel>sell product to third</Rel>
                <Var>z</Var>
            </Atom>
        </prohibition>
    </then>
    <if>
        <And>
            <Atom id="Excl-if-atm1">
                <Rel>is contained in contract</Rel>
```

```
                    <Var>z</Var>
               </Atom>
               <Atom id="Excl-1-if-atm2">
                    <Rel>has contract party</Rel>
                    <Var>x</Var>
               </Atom>
          </And>
     </if>
</Implies>
```

The consequence for violating the exclusivity clause (to pay a penalty fee) is then assumed as a fact:

```
<Atom id="ExclPen">
     <penalty id="ExPn-pnl1">
          <obligation id="obl2" subject="z" beneficiary="x"
          timesBlock="#t2">
               <Atom id="ExPn-pnl1-atm1">
                    <Rel>award of penalty fee to</Rel>
                    <Var>z</Var>
                    <Data> $750 </Data>
               </Atom>
          </obligation>
     </penalty>
</Atom>
```

A new rule that connects the reparation with the violation of the clause, and the reparation with the penalty (see the `penalty="ExPn-pnl1"` attribute) is created. The reparation is triggered only if the subject violated the clause and has paid the penalty fee to the other party.

```
<Implies id="Excl-rep">
     <then>
          <reparation id="Excl-rep1" penalty="ExPn-pnl1"/>
     </then>
     <if>
```

```
            <violation source="#Exclusive"/>

      </if>

</Implies>
```

Reasoning with deontics has not been implemented by the present research as it does not represent a central aspect for the purposes of the present framework, which focuses on judicial subsumptions material circumstances under legal concepts, stretching up to the evaluation of obligations and rights and to the application of legal consequences only to a limited extent. Nevertheless, deontics are needed for proper reasoning and LegalRuleML provides the way to implement them in the present research.

5.7.6. Argumentation Schemes

Considering that the argumentation of the judge is a particular implementation of a selected set rules coming from the law, and that the judge follows a particular schema of argumentation reasoning, three levels are needed:

- To model the rules coming from the law that are valid **erga omnes**;
- To describe the pattern of the judge reasoning that could be reused in other similar cases through **analogy**;
- To model the premises as meta-rules valid only **inter pares** or for the specific case.
- Therefore arguments have to be calculated dynamically on the base of the facts applied to the rules.

The results of the present research suggest the following new elements for distinguish the rule and the arguments:

```
<lrml:hasStatement>

 <lrml:ArgumentStatement key="arg1">

  <lrml:hasTemplate>

   <lrml:Argument         key=":rule1"         schema="&argument-
ontology;#compRegulation">

    <lrml:Conclusion>

     <ruleml:not key=":not1">

     <ruleml:Atom>

      <ruleml:Rel iri="#applies"/>

      <ruleml:Var>X</ruleml:Var>

      <ruleml:Ind>Judge_not_competent </ruleml:Ind>

     </ruleml:Atom>
```

```
          </ruleml:not>
         <lrml:Premises>
          <ruleml:Atom></ruleml:Atom>
          <ruleml:Atom></ruleml:Atom>
          <ruleml:Atom></ruleml:Atom>
         </lrml:Premises>
        </lrml:Argument>
      </lrml:hasTemplate>
     </lrml:ArgumentStatement>
```

[Governatori 2011] demonstrated how it is possible to model the arguments as a particular case of defeasible logic, nevertheless in the context of LegalRuleML, which features a strong distinction between *PrescriptiveStatement* and *ConstitutiveStatement*, it is necessary to introduce a new typology, namely *ArgumentStatement*, for distinguish the arguments from normative and constitutive rules. Secondarily a schema could be applied to the argument using the attribute `schema="&argument-ontology;#compRegulation"`. The arguments have to be maintained at a meta-level that is able to invoke the rules according to the schema adopted by the judge.

5.7.7. A table summarizing the features of LegalRuleML and how they meet the requirements set by the present research

Ontological requirements	**LegalRuleML**
Four-Layers classification of legal documents	LegalRuleML gives account for the four-layer classification based on the FRBR.
Connection to Linked Data	One of the tasks of LegalRuleML is an approach to modelling orientated towards Linked Data not only for the semantics of raw data (acts, contracts, court files, judgements, etc.), but also of rules in conjunction with their functionality and usage. The Italian Senate has made available a SPARQL endpoint for legislative Linked Data[33].

[33] http://dati.senato.it/23.

Isomorphism	Several blocks of LegalRuleML are dedicated to annotate the original legal sources and to connect them to rules, so permitting an N:M relationship also on the authorial level.
Advanced legal concepts representation	LegalRuleML is explicitly built to represent complex effects of legal phenomena, but a tool for NLP or for facts/rules parsing is missing.
The Middle-Out Approach	LegalRuleML does not include the model for automatic case based reasoning. An external module is required.
Description Logics	LegalRuleML does not include an OWL/RDF ontology. It must be therefore built outside, exporting to LegalRuleML. The libraries for this import, however, are missing.
Modularity	RuleML's modular system of schemas for XML permits high-precision web rule interchange.
Syntactical requirements	
Defeasible Logics	LegalRuleML includes full support for defeasible logics.
Dungean Semantics	LegalRuleML do not explicitly model Dungean semantics. Specific rules could simulate it (presuming with a defeasible rule that every statement is true) but the structure of the rules would be far from intuitive.
Deontics	LegalRuleML includes full support for deontics.
Argumentation Schemes	LegalRuleML does not give explicit account for argumentation schemes. Any support for argumentation would thus require an extension of LegalRuleML.
Proof Standards	LegalRuleML does not explicitly model proof standards. [Governatori] explains how to write defeasible rules that represent proof standards.
Factors	LegalRuleML allows including factors into the reasoning process, but it is necessary to extend the reasoner in order to process factors in the same way as [Bench-Capon

	2012].
Dimensions	In order to perform a reasoning on dimensions in judicial cases similar to [Bench-Capon, Representing Popov v. Hayashi with Dimensions and Factors, 2012] the reasoner, given the rules, the facts and the conclusions of reasoning, should produce the (best possible) priority relation between the rules. The reasoner is the one responsible for this, while LegalRuleML seems adaptable to the task.
Values	Being there no argument schemes, it is not possible to introduce values through a specific argument scheme. It is therefore necessary to simulate the role of purpose-based reasoning in a way similar to [CIT] or through modal logics.
Temporal reasoning	LegalRuleML includes an advanced temporal model that can manage combined effects of norms such as multiple efficacy suspensions (see chapter 2).
Procedural argumentation	LegalRuleML does not give explicit account for the argumentation process. It could be extended, exploiting its good temporal model, to identify steps in a sequence and reason on the basis of them.

Chapter 6

A State of the Art of Hybrid Engines

[Das] "Hin- undHerwandern des Blicks zwischen Lebenssachverhalt und Rechtsnormen ist das generell kennzeichnende Merkmal der Rechtsanwendung."

One's attention must shift back and forth between the evidence and legal sources when trying to subsume facts under legal terms.

– Karl Engisch, *Logische Studien zur Gesetzesanwendung*.

"True information does good."

– Julian Assange.

6.1. Introduction

Several of the modelling issues which stand beyond the requirements described in the last chapter can be faced through the so-called *hybrid approach*. In the conception of [Hermann 1997], *hybrid systems* overcome the limitation of rule bases in representing sample data by splitting the rulebase itself into two parts, one formulated by a human expert, and another one generated by machine learning. The acquisition is guided by a fuzzy neural network capable of translating knowledge into rules and of applying linguistic fuzzy variables from fuzzy logic[34].

The hybrid system described in [Hermann 1997] was conceived to automatically acquire case-based knowledge in addition to explanation-based knowledge from an expert. It is meant to be used in medical diagnoses, and its usefulness is demonstrated through an application to the diagnoses of electroencephalograms, proving to be able to outperform the visual diagnosis by a human expert for some phenomena.

In this concluding chapter, two hybrid structures for knowledge management are presented, and their features are analyzed in order to verify which issues (of the ones underlined in the last chapter) they seem apt to resolve.

6.2. SPINdle

[34] Fuzzy logic is not covered by the present research. See [Novák et al. 1999] for a general description of its principles.

6.2.1. Overview

SPINdle is an open-source, defeasible logic engine for business process software which automatically detects compliance levels to complex business rules and governance procedures. Its powerful reasoning engine written in Java computes the consequence of theories in both basic and modal defeasible logics, detecting anomalies. Its reasoning is scalable, and it supports variants of defeasible reasoning such as ambiguity propagation and well-founded semantics. Results from [Lam and Governatori 2009] show that SPINdle can handle inferences with thousands of rules in less than three seconds, increasing almost linearly proportional to the size of the theories tested. It is also possible to optimize those performances and the memory usage through different settings. The software can be used as a standalone theory prover, or be embedded as a defeasible logic rule engine.

6.2.1.1. Ambiguity propagation and well-founded semantics

Before describing the architecture of SPINdle, two preliminary concepts will be explained: that of ambiguity propagation (as counterposed to ambiguity blocking) and that of well-founded semantics.

A literal is *ambiguous* if there is a chain of reasoning that supports the truth of the literal, and another that supports the truth of its negation, and the superiority relation does not resolve this conflict. With the *ambiguity blocking* behavior, neither the literal nor its negation are added to the facts; with the *ambiguity propagation*, both of them are added. While the ambiguity blocking (according to [Stein 1992]) results in unnatural patterns of conclusions, ambiguity propagation results in fewer conclusions being drawn. SPINdle supports both, as either of them can be convenient, depending on the circumstances.

Well-founded semantics was originally developed by [Van Gelder 1991] to provide reasonable interpretation of logic program with negation, and has been applied to extended logic programs and non-monotonic reasoning. It is a skeptical approximation of answer set semantics such that every well-founded consequences of a logic program P is contained in every answer set of P. While some programs are not consistent under answer set semantics, well-founded semantics assigns a coherent meaning to all programs, mainly by removing loops without external support which give birth to unfounded sets.

For more on ambiguity propagation and well-founded semantics, see [Lam and Governatori 2010].

6.2.2. System architecture

The SPINdle architecture consists of three major components which interact with each other in the way shown in Figure 6.1.

- The **I/O Manager** provides an interface to the users in loading the defeasible theories and storing the conclusion after computation. It has two components:
    - the *Theory parser* downloads the theory file, parses the document, and translates it into a data structure that can be processed by the reasoner;
    - the *Theory outputter* stores the modified theory or exports it as an XML string for agent communications;
- The **Theory Normalizer** is responsible for performing the *pre-processing* phase, where a theory is transformed using the techniques described in [Antoniou et al. 2001] into an equivalent theory without superiority relation and defeaters, in order to simplify the reasoning process. More specifically, it performs four tasks:
    - transforming the theory to regular form;
    - emptying the superiority relation;
    - emptying the defeaters;
    - transforming defeasible rules with multiple heads into an equivalent set of rules with single heads;
- The **Inference engine** is responsible for the *conclusions generations* phase which is based, following [Maher 2001] on a series of theory transformations that allow to:
    - assert whether a literal is probable or not, and the strength of its derivation;

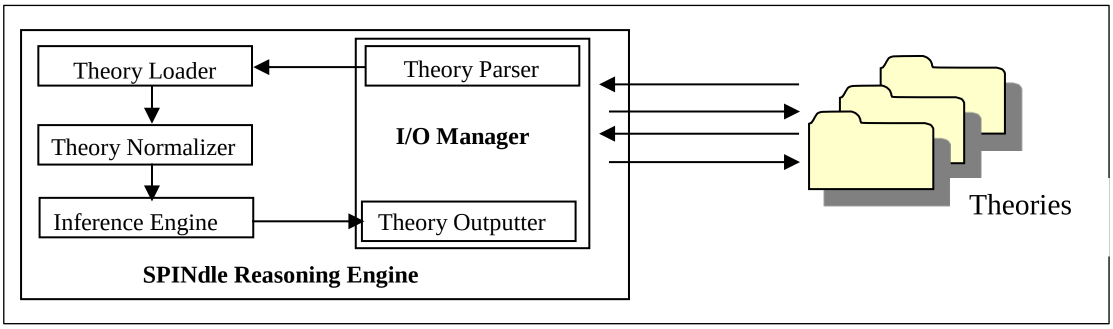


Fig. 6.1 – SPINdle architecture

    - reduce and simplify the theory.

6.2.2.1. Defeasible Theories

Defeasible theories in SPINdle share a common structure which contains the main elements of defeasible logics, namely:

- **Facts**, in form of (modal) states of affairs, describing information in the domain which is known to be true in the form `>>[M]a`, where `[M]` is the model operator representing mental states of the literal `a`;
- **Rules** describing the relations between premise(s) and conclusion in the form `rX[M]:a1,a2,...,aX,-b1,-b2,...,-bX->c1,c2,...,cX`, where `a`, `b`, `c` represent facts (premise, negated premise and conclusion respectively) and `->` represents a strict rule (while a defeasible rule is represented through the symbol `=>`);
- **Defeaters** that prevent some conclusion to occur through the symbol `~>`;
- **Superiority relations** defining priorities among rules in the form `r1>r2`.

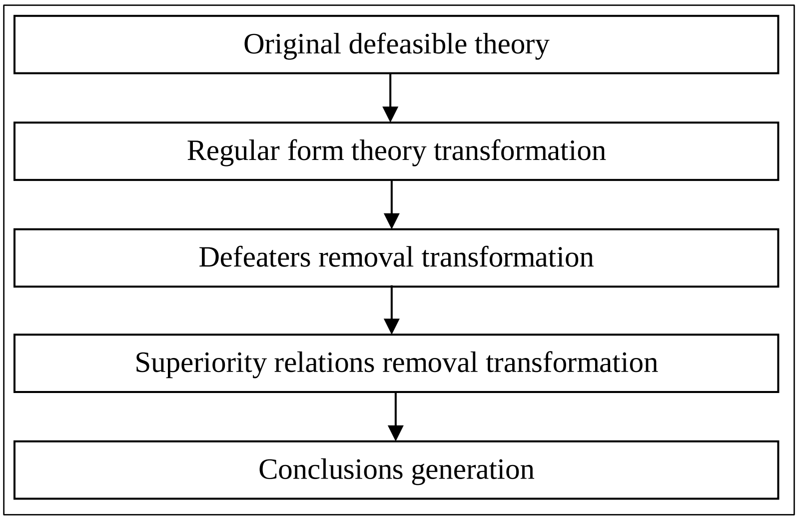


Fig. 6.2 – Defeasible theory inference process in SPINdle

6.2.2.2. The Temporal Model

SPINdle's temporal model is based on the theory described in [Governatori et al. 2005], which is focused on the temporal dynamic treatment of obligations, in connection to the model of directed obligations. The model, however, does not take into account the three-layered structure of time intervals for legal rules (See 5.3.2.) and therefore it is not suited to represent obligations arising from laws: in those cases it is not possible to write the laws *once and for all*, because the reasoner would not be able to properly calculate retroactive effect of reviviscence of repealed law, or the concept of suspension and suspension of the suspending provision.

6.2.2.3. The Inference Process

As already stated in 6.2.1., SPINdle divides the reasoning process into two phases: *pre-processing* and *conclusions generation* (Figure 6.2).

After a defeasible theory has been loaded into the system, SPINdle determines the types of theory normalizer and inference engines to be used based on the elements that appear in the theory. It then applies different types of transformations depending on the requirements of the associated inference approach, or on the user's choice. SPINdle identifies the constraints in the theory that cannot be honored by the inferencing engines, prompting the user for a choice between alternatives. Finally, the reasoner inferences on the theory following two steps for each fact:

- Asserts the fact (an atom) as a conclusion and removes the atom from the rules where the atom occurs positively in the body, and *deactivates* (removes) the rule(s) where the atom occurs negatively in the body. The atom is then removed from the list of atoms;
- Searches for rules with empty head. It takes the head element as an atom and searches for rule(s) with conflicting head. If there are no such rules then the atom is added to the list of facts and the rule will be removed from the theory. Otherwise the atom will be added to the pending conclusion list until all rule(s) with conflicting head can be proved negatively.

On termination (when one of the two steps fails, or when the list of facts is empty), the reasoner outputs the set of conclusions.

Since each fact is processed once, and since for each fact the whole set of rules is scanned, the complexity of the above algorithm is the product of the number of distinct facts for of the number of rules in the theory. This can however be improved through the use of proper data structure (a list for each fact which indicates the rules it occurs in) which reduces the complexity to the product of the number of facts for the maximum number of rules a fact is associated with. Also, the algorithm is a generalized version that is common for both standard defeasible logic and modal defeasible logic. In the case of modal defeasible logic, due to the modal operator conversions, an additional process adding extra rules to the theory is needed. In addition, for a fact with a modal operator, besides its complement, the same literal with modal operator(s) in conflict with the first one should also be included in the conflict literal list, and only the fact with strongest modality will be concluded.

6.2.3. An example of judicial reasoning based on SPINdle

SPINdle can be used as a standalone theory prover or it can be embedded into Java applications as a defeasible rule engine. The present section contains a brief example of how the Judicial Framework of the present research could be represented in the SPINdle rule language, and which are the expected results from its theory prover. Embedding of SPINdle into an application for judicial argumentation, or extending it in one of its various customizable aspects (theory parser, theory outputter, theory normalizer, inference engine) has not been tried by the present research. In order to reach the expressivity of the Carneades System, however, the defeasible rule engine of SPINdle should be enhanced with the concepts of argumentation processes (see Chapter 4).

The following example contains a model of the rules (presented in 4.3.3.) which deals with the application of the effects of a law. Assumed that a circumstance *a* is relevant under the law Art1341co1, rule representing exceptions to the application of that law are modelled.

```
db20:=>efficacious(clause)
db21: contentType(clause), @applies=general, @applies=unilateral, @applies=-knowable =>-efficacious(clause)

db22: contentType(clause), @applies=reproduceslawdisposition ~>-efficacious(clause)

db21>db20

db22>db21
```

If following facts are available:

```
set @applies=general

set @applies=unilateral

set @applies=-knowable

set @applies=reproduceslawdisposition

>>contentType(clause)
```

The reasoner should return the following conclusions, where `+D` means definitely provable, `-D` means not definitely provable, `+d` means defeasibly provable, and `-d` means not defeasibly provable:

```
+D contentType(clause)

-D efficacious(clause)

-D -efficacious(clause)

+d contentType(clause)

+d efficacious(clause)

-d -efficacious(clause)
```

From this application, it can be seen that the priority relation allows to build very straightforwardly when dealing with exceptions (avoiding the `RelevantExArt` Intermediate class, described in 4.3.3.). The approach of a relative ranking rather than an absolute "strength" of rules has however some backsides: particularly in the judicial settings, where the hierarchy of sources is not always clear and contradicting positions from the very same institution may exist (it is the case of the Court of Cassation in Italy, which often keeps a contradictory stance on a topic for several years, until it reaches a uniformity of judgement among all its sections), adopting a logics that forces to resolve contradiction may prove counterproductive. Thus, it may be convenient to use this approach when dealing with laws, but when dealing with legal interpretations and legal sources such as case-law and jurisprudence it is necessary to represents the conclusions brought forward by the rules also when they contradict each other. Considering that the intention is to rely on OWL databases, the need to represent intermediate classes (`RelevantExArt`) does not disappear by adopting this way of resolving inconsistencies. It thus seems the case to rely on absolute values, not using superiority relations at all.

6.2.4. An Evaluation of SPINdle under the ontological and syntactical requirements set by the present research

Following is a table summarizing how the features of SPINdle meet the requirements set in 5.5.:

Ontological requirements	**SPINdle**
Four-Layers classification of legal documents	The rule language could be extended to include such information. Modal components of the defeasible model may be exploited in that direction.
Connection to Linked Data	The I/O manager could be extended to retrieve/share knowledge.
Isomorphism	The rule language and I/O manager could be extended to ensure a strict link between the text and the data.
Advanced legal concepts representation	Modal defeasible logics allow complex constructs for the representation of the dynamics of legal concepts.
Middle-Out Approach	The theory parser can be extended to read OWL ontologies.
Description Logics	Modal component of the logical model ensure the conservation of information from description logics.
Modularity	The I/O manager does not load multiple theories, so the modules which are relevant for the reasoning have to be selected and packed into a single theory outside of SPINdle.
Syntactical requirements	
Defeasible Logics	SPINdle has full support for all rule types of defeasible logics.

Dungean Semantics	SPINdle does not explicitly model Dungean semantics. Specific rules could simulate it (presuming with a defeasible rule that every statement is true) but the resulting structure of the rules would be far from intuitive.
Deontics	Support for modal defeasible logics allows the representation of deontic concepts and Hohfeldian relations with good accuracy.
Argumentation Schemes	SPINdle does not explicitly model argumentation schemes. [Governatori 2011] argues that its model is sufficient to represent argumentation schemes (see chapter 4.6.).
Proof Standards	SPINdle does not explicitly model proof standards. [Governatori 2011] explains how to write rules for SPINdle that represent proof standards.
Factors	The rule language allows to include factors into the reasoning process, but it is necessary to extend the reasoner in order to process factors in the same way as [Bench-Capon 2012].
Dimensions	In order to perform a reasoning on dimensions in judicial cases similar to [Bench-Capon 2012] the reasoner, given the rules, the facts and the conclusions of reasoning, should produce the (best possible) priority relation between the rules. This would require a drastic customization of the reasoning engine.
Values	Being there no argument schemes, it is not possible to introduce values through a specific argument scheme. It is therefore necessary to simulate the role of purpose-based reasoning in a way similar to [Bench-Capon and Sartor 2003] or through modal logics.
Temporal reasoning	SPINdle gives account for time and time intervals, but its basic model does not include three-layered structure for force/efficacy/application times of legal norms. The rule language and the reasoner should be extended in that direction.
Procedural argumentation	The SPINdle model seems to lack the proper structures to manage a turn-based argumentation process, simulating multiple stages of reasoning where each stage's task is determined by the precedent stages.

6.3. Drools

6.3.1. Overview

Drools is a business rule management system. Its hybrid rules engine started as a production rule system based on the Rete algorithm (see [Forgy 1982]), until version 5.x introduced Prolog style backward chaining reasoning as well as some functional programming styles. The Drools Rete implementation is called *ReteOO*, signifying that Drools has an enhanced and optimized implementation of the Rete algorithm for object oriented systems. It combines Object Oriented (OO) Paradigm entities with rules in order to let them interact in a transparent way. The rule engine is nested into a business logic integration platform, which combines the rules with workflow and event processing tools to create an enterprise framework for the construction, maintenance, and enforcement of business policies in an organization, application, or service.

6.3.1.1. Drools Expert

The Drools Expert module is a declarative, rule based coding environment. A Drools rule has the following structure:

```
rule "name"

  attributes

    when

      condition

    then

      action
```

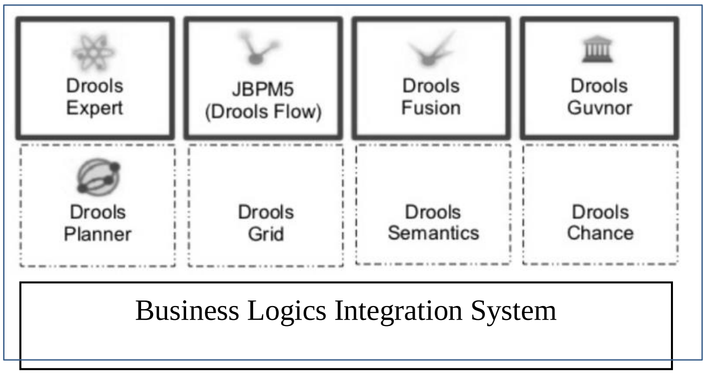


Fig. 6.3 the components of the Drools system.

It is not a classical conditional structure if-then of propositional logic (see 1.2.2.1.). Instead, the when-then model is adopted. Using the Rete algorithm, Drools executes the conclusions whenever the patterns in the conditions (which follow a precise syntax) are matched by a fact. The Expert module contains decision tables, rule templates, a guided editor and ruleflow authoring tools.

6.3.1.2. jBPM

jBPM is a *Business Process Management* (BPM) Suite, allowing to model business goals by representing in a flow chart the steps that need to be executed to achieve that goal. The core of jBPM is a light-weight, extensible workflow engine written in pure Java that allows the execution of business processes using the latest BPMN 2.0 specification, a standardized specification that defines a visualization and XML serialization of business processes and can be extended to include more advanced features. It can run in any Java environment, embedded in an application or as a service.

6.3.1.3. Drools Fusion

The most interesting part for the present research, even if it handles concepts (such as agents and event-based reasoning) which have barely been touched by the present thesis, is Drools Fusion, the module responsible for enabling event processing capabilities. Complex Event

Processing, or CEP, is a kind event processing that deals with the task of processing multiple events with the goal of identifying the meaningful events within the event cloud. CEP employs techniques such as detection of complex patterns of many events, event correlation and abstraction, event hierarchies, and relationships between events such as causality, membership, and timing, and event-driven processes

To achieve the flexibility and power of behavioural modelling (see [Rapisarda and Willems 2006]), a platform must understand Rules, or Processes, or Events as primary concepts and allow them to leverage on each other strengths, without taking the perspective of only one of them.

Drools Fusion, in this scenario, is an independent module, but still completely integrated with the rest of the platform, that adds a set of features to enable it to:

- **Handle events as first class citizens**: events, in Drools 5.x, are a special entity that represent a record of a significant change of state in the application domain. They have several unique and distinguishing characteristics, like being usually immutable, having strong temporal constraints and relationships. Drools Fusion understands events by what they are and allows users to model business rules, queries and processes depending on the presence or absence of them;

- **Support advanced event management**: Drools Fusion supports both streams and clouds of events. In case of streams it supports asynchronous, multi-thread feeding of events. It also supports calculations on moving windows of interest, be it temporal or length-based windows. Finally, it is able to identify the events that are no longer needed and dispose them as a way of freeing resources and scaling well on data volumes.

- **Support reasoning over absence of events**: Drools Fusion leverages on the capabilities of the Drools Expert engine, allowing complete and flexible reasoning over the absence of events, including the transparent delaying of rules in case of events that require a waiting period before firing the absence.

- **Support for temporal reasoning**: Drools Fusion adds a set of temporal operators for modelling and reasoning over temporal relationships between events (see below 6.3.2.1.).

6.3.1.4. Drools Guvnor

Drools Guvnor is a centralized repository for Drools Knowledge Bases, including web based GUIs, editors, and tools to store versions of rules, models, functions, processes related to the KB. Access is controlled, and it is possible to lock down access and restrict features so domain experts (not programmers) can view and edit rules without being exposed to all the features at once.

6.3.2. A Drools-based rule language for the legal domain

In [Palmirani et al. 2012] an inference engine based on Drools ver. 5.4 is presented. The engine relies on a hybrid technology as described in [Bragaglia et al. 2010] and [Sottara et al. 2010], and uses LegalRuleML (see 5.6.)as rule language. Its aim is to sustain expert systems able to take into consideration all the events occurred during a given time lapse and to match them to appropriate conclusions by means of a rulebase. For the legal domain, in fact, the best option is to build an expert system focalized on concrete applications, which is also the reason why Drools seems a valid candidate for the job. It needs, however, several extensions in order to properly manage legal knowledge. It needs in particular:

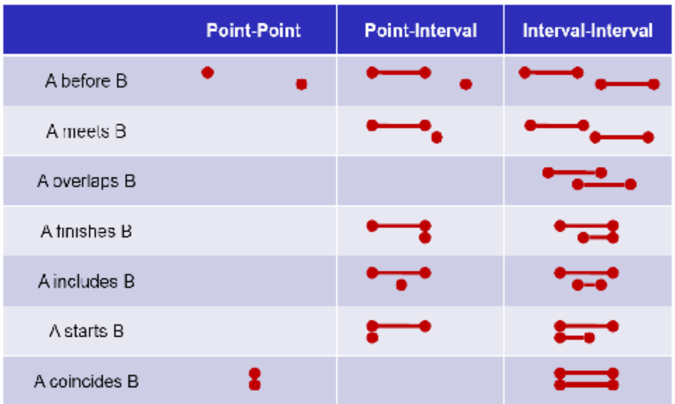


Fig. 6.4 – Allen's temporal model

– a complex **temporal model** capable of managing retroactivity (see next section);

– a module to effectively **import/export** LegalRuleML files;

– a more abstract model for **defeasibility**;

– a refined front-end **interface** for the rule viewer integrated with Drools.

6.3.2.1. The temporal model

The model used by Drools to perform temporal reasoning is the model of [Allen 1983] (see Figure 6.4). This model is effective only then dealing with business-like rules, which refer to events which live in the present and die in the future. On the contrary, dealing with legal events and legal rules involves a complex mix of past, present and future to deal with: for example, a modification may occur in the present but have a retroactive effects. For this reason the temporal attributes implemente by Drools to manage the lifecycles of events and rules are not enough to represent the legal domain. In [Palmirani et al. 2012] this limitation had been partially supplied by adding temporal constraints as patterns to be matched: this was done by introducing events as instances, with a precise timestamp and duration. These temporal parameters can be added to every rule, or as metarules to control the rule flow with jBPM. It is possible, for example, to force the rule engine to activate only the rules that match certain temporal parameters, allowing to effectively deal with rule versioning. This kind of rule flow management, however, cannot deal effectively with multiple versions of legal rules which are applicable in the same time. In this case it is possible to use the construct *salience* to rank the rules, but this ranking has to be explicit and preemptive: it cannot be derived through a function, and has to be manually set for all the rules.

6.3.2.2. Current state of the project

An example of legal rules written in LegalRuleML language is contained in the Appendix. In the current state, it is not possible to test them within a proper reasoner, since the extension of Drools suggested in [Palmirani et al. 2012] has not been realized yet. It would be interesting to verify how this new model supports the temporal dimensions, and how it supports arguments. In fact, as already said in Chapter 5, if this new rule language seems promising under the aspect of time management, it lacks proper instruments to manage the argumentation process.

6.3.3. An Evaluation of a Drools/LegalRuleML-based rule language under the ontological and syntactical requirements set by the present research

Complexively, Drools has similarities with the IMPACT Policy Modeling tool based on Carneades 2.x (see http://ec.europa.eu/information_society/apps/projects/logos/8/247228/080/deliverables/001_IMPACTD42.pdf): in particular, both the systems are intended to cover the gap between complex abstract rules, written by IT specialists, and practical use cases, to be managed by domain experts. Both rely on defeasible rules to achieve this goal, and are aimed at producing evaluations and suggestions to the user, mostly through the visualization tool. The main difference between the two program suites is that the IMPACT Policy Modeling tool uses argument schemes while Drools uses rules templates, the latter not being capable of influencing the concept burden of proof (significatively enough, burden of proof itself does not seem to be as hard-coded in Drools as it is in the project from IMPACT). However, the advanced conception of rules, processes and events brought forward by the Drools Fusion module seems to hold a great potentiality for the representation of argumentation as a process. Following is a table summarizing how the features of Drools (in the implementation which features LegalRuleML for the rule language and Akoma Ntoso for the metadata) meet the requirements set in 5.5.:

Ontological requirements	**Akoma Ntoso+LegalRuleML Drools**
Four-Layers classification of legal documents	The Akoma Ntoso standard explicitly supports this.
Connection to Linked Data	One of the tasks of LegalRuleML is an approach to modelling orientated towards Linked Data not only for the semantics of raw data (acts, contracts, court files, judgements, etc.), but also of rules in conjunction with their functionality and usage. The Italian Senate has made available a SPARQL endpoint for legislative Linked Data[35].
Isomorphism	Several blocks of LegalRuleML are dedicated to annotate the original legal sources and to connect them to rules, so permitting an N:M relationship also on the authorial level.

[35] http://dati.senato.it/23.

Advanced legal concepts representation	Akoma Ntoso and LegalRuleML are explicitly built to represent complex effects of legal phenomena, but a tool for NLP or for facts/rules parsing is missing.
The Middle-Out Approach	Akoma Ntoso is able to provide semantically rich metadata for bottom-up ontology and rules construction, but in order to introduce a top-down core ontology it is necessary to build an ontology outside of the system, and Drools can be configured to reason on both the Akoma Ntoso metadata and the ontology set.
Description Logics	Akoma Ntoso relies on XML, but does not include an OWL/RDF ontology. The ontology must be therefore built outside, importing from Akoma Ntoso and exporting to LegalRuleML.
Modularity	RuleML's modular system of schemas for XML permits high-precision web rule interchange.
Syntactical requirements	
Defeasible Logics	LegalRuleML includes full support for defeasible logics (see 5.6.2.).
Dungean Semantics	Nor LegalRuleML or Drools explicitly model Dungean semantics. Specific rules could simulate it (presuming with a defeasible rule that every statement is true) but the structure of the rules would be far from intuitive. A better investigation of Drools' features could uncover the possibility to represent dungean semantics through Drools Expert.
Deontics	LegalRuleML includes full support for deontic logics (see 5.6.5.).
Argumentation Schemes	LegalRuleML does not give explicit account for argumentation schemes. Any support for argumentation would thus require an extension of LegalRuleML, Drools Fusion and jBPM. A possible extension is presented in 5.6.6.
Proof Standards	LegalRuleML does not explicitly model proof standards. [Governatori 2011] explains how to write defeasible rules

	that represent proof standards.
Factors	The rule language allows to include factors into the reasoning process, but it is necessary to extend the reasoner in order to process factors in the same way as [Bench-Capon 2012].
Dimensions	In order to perform a reasoning on dimensions in judicial cases similar to [Bench-Capon 2012] the reasoner, given the rules, the facts and the conclusions of reasoning, should produce the (best possible) priority relation between the rules. Drools should be extended in order to achieve this.
Values	Being there no argument schemes, it is not possible to introduce values through a specific argument scheme. It is therefore necessary to simulate the role of purpose-based reasoning in a way similar to [Bench-Capon and Sartor 2003] or through modal logics.
Temporal reasoning	LegalRuleML includes an advanced temporal model that can manage combined effects of norms such as multiple efficacy suspensions (see 5.3.2).
Procedural argumentation	LegalRuleML does not give explicit account for the argumentation process. Drools' advanced management of events as first-class citizen could be exploited to implement an interconnection between steps of a process and tasks of the reasoning.

6.4. Conclusions

Reasoning on the semantics of judgements is different, complex, and extremely useful. As this thesis demonstrates, it takes the best of the state-of-the art in knowledge representation and argument reconstruction to reproduce the semantics of judgements and doctrine. A consistent part of the results, however, has been already achieved by the giants before us, and our role as dwarfs on their shoulders is to reach an agreement on the required standard on the rule and logics layer of the Semantic Web in the legal field. LegalRuleML constitutes an ideal starting point on this behalf, while reasoning requires the features of modal defeasible logics with an account for modern argumentation theory and complex temporal representation of documents, something which has not yet been achieved by a single application. The central result of the present research is thus the set of requirements contained in 5.5., as they constitute a possible starting point for a new race of AI & Law community towards the representation of legal reasoning with Information Technologies.

The research showed that it is possible not only to create the argumentation diagrams in the style of Wigmore & Toulmin, and to conduct reasoning on them (which was already achieved by Araucaria), but also to create them automatically, importing semantics from a case-based ontology. The feature of Carneades of *argument from ontology* had never been applied to such a concrete and complex sample.

Finally, the present research would like to constitute a call for the constitution of a wide international community composed of lawyers, IT experts, judges and legal professionals, because there is (still?) no way of representing knowledge so complex as legal knowledge without relying on the most complex processors of the universe, namely the human brain. And legal information is so vast, and ever-changing, that a single (or a limited number of) brains will never be able to build a comprehensive jurisprudence of legislation and case-law, nor to keep the pace of its modifications. For these reasons, this work is concluded by citing a phrase contained in 5.5.2.2.:

> The commitment to the community, and the mix of advantages and duties it entails, is not a mere modelling choice, nor a strategy or an approach: it is, evidently, a philosophical shift in the concept of scientific production in our age.

6.5. Further Work

6.5.1. The Judicial Assistant Tool

On the basis of the ontology set described in Chapter 3 and of the argumentation system described in Chapter 4 a software could be built, which manages judicial sentences enriching the knowledge contained in them in the following ways:

- Finding relevant precedents that are not explicitly cited in the decision;
- Validating the adjudications of the judge on the claims brought forward by the parties during the trial on the basis of applicable rules, accepted evidence, and interpretation;
- Suggesting legal rules/precedents/circumstances that could bring to a different adjudication of the claim.

Advanced features of the Judicial Assistant tool may include:

- **Compliance check** of contract drafts, i.e. through a plugin of a word processor using NLP techniques to recognize sentences and clauses that could be relevant under consumer law;
- **Juridical analysis tools** for legal professionals, enriching case-law collections by semantically relating and grouping precedents for lawyers to browse, making the precedent extraction process for legal cases easier and more effective;
- **Judgement management tools** for courts and tribunals, useful to evaluate and optimize judgements (i.e. integrated into a word processor to help the judge while writing the judgement, avoiding grounds for appeals due to missing elements in the decision's groundings);

- **Impact analysis tools** for legislators, providing a list of (common or uncommon) judicial interpretations for a given law, in order to take them into account when modifying that law;

- **New tools** representing formalized legal doctrine and case law, where legal experts could rely on a social platform to share their views and interpretations on a law or a precedent, using a graphical interface and a formal argumentation structure instead of plain text.

6.5.2. Generalization of Patterns

In order to test and refine the set of ontologies built from the Italian consumer contracts-related case law (or at least its core part) a set of foreign sentences in the same field could be analyzed, in order to verify the differences between the meanings of the main legal concepts and the possibility of representing those differences in the ontologies. This is an important topic in European Law and in the perspective of the growing phenomenon of *international citation*. By developing a multi-lingual, multi-system judicial ontology it would be possible to study and compare law principles under an international perspective. The same research can be conducted on argumentation patterns: recurrency of argumentation patterns in different legal systems could be compared and connections with differences in law principles or specific discipline could be studied.

6.5.3. International judicial taxonomy

The research could further develop in the direction of multi-lingual approach to legal ontologies by trying to create better semantic connections between the legal text and the rules represented in it. With help from lexicon-related projects, such as FrameNet (see 5.3.3.3.) and *Legal Taxonomy Syllabus* of the University of Turin (see [Ajani et al. 2008]), the research can try to *fill the gap* between the formalization of language and the representation of legal principles. It will be probably necessary to identify two layers in the ontology set, transversal to the core and domain components: one more connected to the linguistic and lexicon part, and another linking the legal concepts. In this way, the research could study the differences between languages in representing legal concepts and judicial argumentation.

6.6. Personal Remarks

As G. W. F. Hegel writes in the opening quote of this thesis (0.), “World history is a court of judgement”: it is there that the correspondence between the ideas and the facts is verified. Following Hegel: “Truth in philosophy means that concept and external reality correspond”. The kind of *judgement* Hegel ascribes to *World history* is hence a *quest for Truth*, rather than a *quest for Justice*.

The object of this thesis is strictly connected to logics, which tells about the concept of Truth and how to get as close as possible to it, no matter if we can never seize it. In a world where the suffering of many depends on the manipulation of this precious and fragile concept, any *quest for Truth* is to be lived and fostered as a quest for making something better of this place. The final goal of Research should thus be that of shredding the clouds which cloak our perceptions: in the teaching of the great philosopher Hannah Arendt it is in those clouds that the very roots of contemporary barbarism lie.

This thesis, apparently, also buzzes around a concept which is equally difficult to seize, to the point that in a perfect World it would have no meaning: the concept of Justice. [Kelsen 1957] defines the longing for justice as men's eternal longing for happiness: it is happiness that man cannot find alone, as an isolated individual, and hence seeks in society. It is, together with Freedom, one of the hallowed concepts of our times, but any *quest for Justice* is a dangerous enterprise: different conceptions of Justice are, in fact, the origin of wars between peoples[36]. Moreover, zeal can hide the most human, desirable and powerful feeling: compassion. Western *civilization* has so much to learn on this behalf!

Chomsky fairly interprets *World history* as a *court of judgement* of our times in his lecture *Government of the Future*:

> “We have today the technical and material resources to meet man’s animal needs. We have not developed the cultural and moral resources or the democratic forms of social organization that make possible the humane and rational use of our material wealth and power. Conceivably, the classical liberal ideals, as expressed and developed in their libertarian socialist form, are achievable. But if so, only by a popular revolutionary movement, rooted in wide strata of the population, and committed to the elimination of repressive and authoritarian institutions, state and private. To create such a movement is the challenge we face and must meet if there is to be an escape from contemporary barbarism.” [Chomsky 2005]

“Democratic forms of social organization that make possible the humane and rational use of our material wealth and power”, this is what we should all be focusing on now. Not Finance.

[36] On the contrary, wars waged by *ruling elites* require much less.

6.7. Acknowledgements

This Ph.D thesis was mostly conceived, and written, in Italy. My first thought goes therefore to all the Italians which, across time and space, taught me through images and words how to stay human. How beauty survives in the mud. How competence, passion and patience can flourish among corruption and emptiness. While trying to endure my thought goes to Vittorio Arrigoni (1975-2011), a nonviolent paladin against barbarism. One of the few just men of those troubled times.

I would have never made it to the end of this work without the constant help and support from Prof. Monica Palmirani, severe and forgiving, a model of perseverance and strength. The research periods spent in Berlin were wonderful and challenging thanks to Prof. Thomas F. Gordon, a true gentleman and a great teacher, together with his assistants Jurma and Pierre. I miss the sincere and intense confrontations the four of us had on the most diverse topics.

The rest of the time I was complicating the life of my colleagues at CIRSFID: Corrado, Erica, Raffaella, Giuseppe, Silvia, Filippo, Migle, Tazia, Simona, Patrizio, Alberto. Each of them easened my work (or saved me from troubles) more than a couple of times: I thank them all, and I remember the pleasant atmosphere they created in that dusty and creaky *noble floor* of Palazzo Dal Monte Gaudenzi.

It would have been unconceivable even to start such a work if I wasn't grown by outstanding parents such as mine, my mother Candida teaching me how to struggle for a dream, my father Angelo how to cherish it. Beside them my sister Federica, giving me a sample of how to be a free, open-minded human being.

The worst burden of realizing this thesis – the stress and the annoying man it turns me into – fell on the back of Elisabetta, my ally and the cornerstone of my life. I can never pay her back for being the hardest spirit I ever met.

This thesis is dedicated to Sebastiano Sunny Giannandrea,
World's latest user,
harbinger of merriness.

Bibliography

[Abate and Jewell 2001] Jewell, E. J., Abate, F.: *The new Oxford American Dictionary*. Oxford University Press, New York, 2001.

[Ajani 2008] Ajani, G., Boella G., Lesmo L., Mazzei A., Radicioni D. P., Rossi P.: *Legal Taxonomy Syllabus: Handling Multilevel Legal Ontologies*. In: Proceedings of Langtech 2008.

[Alchourron 1996] Alchourrón, C. E.: *On law and logic*. In: Ratio Juris 9, 1996, 331–48.

[Alchourron and Bulygin 1971] Alchourrón, C. E., Bulygin, E.: *Normative systems*. Vol. 5. Vienna, Springer, 1971.

[Aleven 1997] Aleven V.: *Teaching Case-Based Argumentation Through a Model and Examples*. PhD Dissertation, University of Pittsburgh, 1997.

[Aleven 2003] Aleven, K. D.: Using Background Knowledge in Case-Based Legal Reasoning: A Computational Model and an Intelligent Learning Environment. In: Artificial Intelligence 150, 2003, 183-237.

[Aleven and Ashley 1994] Ashley, K. D., Aleven, V.: *A logical representation for relevance criteria*. In: Topics in Case-Based Reasoning, Springer, Berlin Heidelberg, 1994, pp. 338-352.

[Alexy 1985] Alexy, R.:. *Theorie der Grundrechte*. Suhrkamp, Frankfurt am Main, 1985.

[Alexy 1989] Alexy, R.: *A Theory of Legal Argumentation.* Oxford University Press, New York, 1989.

[Allen 1983] Allen, J. F.: *Maintaining knowledge about temporal intervals*. In: Communications of the ACM, 26(11), 1983, 832-843.

[Antoniou et al. 2001] Antoniou, G., Billington, D., Governatori, G., Maher, M. J.: *Representation results for defeasible logic*. In: ACM Transactions on Computational Logic 2(2), 2001, 255–286.

[Antoniou et al. 2009] Antoniou, G., Dimaresis, N., Governatori, G.: *A Modal and Deontic Defeasible Reasoning System for Modellin Policies and Multi-Agent Systems*. In: Expert Systems With Applications, 2009, pp. 4125-4134.

[Artz and Gil 2007] Artz, D., Gil, Y.: *A Survey of Trust in Computer Science and the Semantic Web*. In: Journal of Web Semantics: Science, Services and Agents on the World Wide Web, 2007. Available at: http://www.isi.edu/~gil/papers/jws-trust-07.pdf.

[Ashley 1990] Ashley, K. D.: Modeling Legal Argument: Reasoning with Cases and Hypotheticals. MIT Press, Cambridge, MA, 1990.

[Ashley 1991] Ashley, K. (1991, 34). Reasoning with cases and hypotheticals in HYPO. *Int. J. Man - Machine Studies*, 753-796.

[Ashley 2009] Ashley, K. D.: *Ontological requirements for analogical, teleological, and hypothetical legal reasoning*. In: Proceedings of the 12th International Conference on Artificial Intelligence and Law, New York, 2009.

[Atkinson 2005] Atkinson, K.: What Should We Do?: Computational Representation of Persuasive Argument in Practical Reasoning. PhD thesis, Department of Computer Science, University of Liverpool, 2005

[Atkinson 2006] Atkinson, K.: *Value-based argumentation for democratic decision support*. In: Frontiers in Artificial Intelligence and Applications 144, 2006, 47.

[Baader and Nutt 2003] Baader, F, Nutt, W.: *Basic Description Logics*. In: Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P. F. (eds): The Description Logic Handbook: Theory, Implementation and Applications, Cambridge University Press, 2003.

[Baader et al. 1994] Baader, F., Franconi, E., Hollunder, B., Nebel, B., Profitlich, H. J.: *An empirical analysis of optimization techniques for terminological representation systems or: making KRIS get a move on*. In: Applied Artifical Intelligence, Special Issue on Knowledge Base Management, 1994, 4:109-132.

[Baker et al. 1998] Baker, C., Fillmore, C., Lowe, J. B.: *The Berkeley FrameNet Project*. In: Proceedings of the 17th International Conference on Computational Linguistics, August 10–14, Montreal, 1998.

[Ballnat and Gordon 2010] Ballnat, S., Gordon, T. F.: Goal Selection in Argumentation Processes - A Formal Model of Abduction in Argument Evaluation Structures. In: Proceedings of COMMA 2010, IOS

Press, Amsterdam 2010, 51-62.

[Ballweg and Seibert 1982] Ballweg, O., Seibert, T. (eds): *Rhetorische Rechtslehre*. Alber, Freiburg, 1982.

[Barabucci et al. 2010a] Barabucci, G., Cervone, L., Palmirani, M., Peroni, S., Vitali, F.: *Multi-layer markup and ontological structures in Akoma Ntoso*. In: AI Approaches to the Complexity of Legal Systems: Complex Systems, the Semantic Web, Ontologies, Argumentation, and Dialogue. Springer, Berlin Heidelberg, 2010, 133-149.

[Barabucci et al. 2010b] Barabucci, G., Cervone, L., Di Iorio, A., Palmirani, M., Peroni, S., Vitali, F.: *Managing semantics in XML vocabularies: an experience in the legal and legislative domain*, Balisage, 2010.

[Beardsley 1950] Beardsley, M. C.: *Practical Logic*. Prentice-Hall, New York, 1950.

[Bench-Capon 2012] Bench-Capon, T. J. M.: *Representing Popov v. Hayashi with Dimensions and Factors*. In: Artificial Intelligence and Law 20, 2012, 15-35.

[Bench-Capon and Coenen 1992] Bench-Capon, T. J. M., Coenen, F. P.: *Isomorphism and legal knowledge based systems*. In: Artificial Intelligence and Law, 1, 1992, 65-86.

[Bench-Capon and Gordon 2009] Bench-Capon, T. J. M., Gordon, T. F.: *Isomorphism and argumentation*. In: Proceedings of the 12th International Conference on Artificial Intelligence and Law, ACM, 2009, 11-20.

[Bench-Capon and Prakken 2010] Bench-Capon, T. J. M., Prakken, H., *Using argument schemes for hypothetical reasoning in law*. In: Artificial Intelligence and Law 18(2), 2010, 153-174.

[Bench-Capon and Rissland 2001] Bench-Capon, T. J. M., Rissland, E. L.: Back to the Future: Dimensions revisited. In: Verheij, B., Lodder, A. R., Loui, R. P., Muntjewerff, A.J. (eds): Legal Knowledge and Information Systems. JURIX 2001: The Fourteenth Annual Conference. IOS Press, Amsterdam, 2001, 41-52.

[Bench-Capon and Sartor 2003] Bench-Capon, T. J. M., Sartor, G.: *A Model of Legal Reasoning with Cases Incorporating Theories and Values*. In: Artificial Intelligence 150, 2003, 97-143.

[Berners-Lee 2009] Berners-Lee, T.: *Tim Berners-Lee on the next web*, March 2009. Retrieved 4/5/2013 from TED, http://www.ted.com/talks/tim_ berners_lee_on_the_next_web.html.

[Berners-Lee et al. 2006] Berners-Lee, T., Hall, W., Hendler, J. A., O'Hara, K., Shadbolt, N., Weitzner, D. J.: *A Framework for Web Science*. In: Foundations and Trends in Web Science, Vol. 1 No. 1, 2006, 1-130.

[Bex 2009] Bex, F.: Evidence for a good story. A hybrid theory of arguments, stories and criminal evidence. Doctoral dissertation, Faculty of Law, University of Groningen, 2009.

[Bex et. al. 2003] Bex, F., Prakken, H., Reed, C., Walton, D.: *Towards a formal account of reasoning about evidence: Argumentation schemes and generalizations*. In: Artificial Intelligence and Law 11, 2003 125-165.

[Bing 1982] Bing, J.: *Uncertainty, decisions and information systems*. In: Ciampi, C. (ed): Artificial Intelligence and Legal Information Systems. North-Holland, 1982

[Bizer et al. 2009] Bizer, C., Lehmann, J., Kobilarof, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S.: *DBpedia - A crystallization point for the Web of Data*. In: Web Semantics: Science, Services and Agents on the World Wide Web, 2009, 154-165.

[Boella et al. 2010] Boella, G., Governatori, G., Rotolo, A., van der Torre, L.: *A Logical Understanding of Legal Interpretation*. In: Proceedings of KR 2010.

[Boer 2000] Boer, A.: *The Consultancy Game*. In: Breuker, J. A., Leenes, R., Winkels, R. G. F. (eds): Legal Knowledge and Information Systems. Jurix 2000: The Thirteenth Annual Conference. Frontiers in Artificial Intelligence and Applications, IOS Press, Amsterdam, 2000, 99-112.

[Boer et al. 2007] Boer, A., Di Bello, M., van den Berg, K., Gordon, T., Förhécz, A., Vas, R.: *Specification of the Legal Knowledge Interchange Format*. Deliverable 1.1., 2007.

[Boer et al. 2008] Boer, A., Winkels, R., Vitali, F.: *MetaLex XML and the Legal Knowledge Interchange Format*. In: Casanovas, P., Sartor, G., Casellas, N., Rubino, R. (eds): Computable Models of the Law, Springer, Heidelberg, 2008, 21-41.

[Boer et al. 2009] Boer, A., Hoekstra, R., de Maat, E., Hupkes, E., Vitali, F., Palmirani, M., Rátai, B.: *CEN Metalex Workshop Agreement*. 2009-08-28 proposal, http://www.metalex.eu/WA/proposal.

[Boley 2006] Boley ,H.: *The RuleML family of web rule languages*. In: Alferes, J. J., Bailey, J., May, W., Schwertel, U. (eds): PPSWR, volume 4187 of Lecture Notes in Computer Science, Springer, Berlin Heidelberg, 2006, 1-17.

[Boley and Tabet 2000] Boley, H. Tabet, S.: *RuleML: The ruleml standardization initiative*, http://www.ruleml.org/, 2000.

[Bondarenko et al. 1997] Bondarenko, A., Dung, P. M., Kowalski, R. A., Toni, F.: *An abstract, argumentation-theoretic approach to default reasoning*. In: Artificial intelligence 93, 1/1997, 63-101.

[Boole 1847] Boole, G.: *Mathematical Analysis of Logic*, Macmillan, Cambridge, 1847.

[Bozzo-Rey 2009] Bozzo-Rey, M.: Jeremy Bentham's Legal and Political Philosophy: Acting Logically to Act Ethically? In: Ethics in Public Life, Salzburg, 2009.

[Brachman 1977] Brachman, R. J.: *What´s in a concept: Structural Foundations for semantic networks*. In: International Journal of Man-Machine Studies, 9(2), 1977, 127-152.

[Brachman 1978] Brachman, R. J.: *Structured inheritance networks*. In: Woods, W. A., Brachman, R. J. (eds): Research in Natural Language Understanding, Quarterly Progress Report No. 1, BBN Report No. 3742, Bolt, Beranek and Newman Inc., Cambridge, MA, 1978, 36-78.

[Brachman and Schmolze,1985] Brachman, R. J., Schmolze, J. G.: *An Overview of the KL-ONE knowledge representation system*. In. Cognitive Science, 9 (2), 1985, 171-216.

[Bragaglia et al. 2010] Bragaglia, S., Chesani, F., Ciampolini, A., Mello, P., Montali, M., Sottara, D.: *An hybrid architecture integrating forward rules with fuzzy ontological reasoning*. In: Hybrid Artificial Intelligence Systems, Springer, Berlin Heidelberg, 2010, 438-445.

[Branting 1991] Branting, L. K.: *Reasoning with Portions of Precedents*. In: Proceedings of the Third International Conference on Artificial Intelligence and Law, Oxford, England, June 25-28, 1991.

[Branting 1999] Branting, L. K.: *A Generative Model of Narrative Cases*. In: Proceedings of the Seventh International Conference on Artificial Intelligence & Law (ICAIL-99), Oslo, Norway, July 14-17, 1999.

[Branting 2000] Branting, L. K.: Reasoning with Rules and Precedents: A Computational Model of Legal Analysis, Kluwer Academic Publishers, Dordrecht, 2000.

[Breuker and Hoekstra 2004] Breuker, J., Hoekstra, R.: *Core concepts of law: taking common-sense seriously*. In: Proceedings of Formal Ontologies in Information Systems (FOIS-2004). IOS-Press, 2004

[Breuker and Van de Velde 1994] Breuker, J., Van De Velde, W. (eds): *CommonKADS Library for Expertise Modeling: reusable problem solving components*. IOSPress/Ohmsha, Amsterdam/Tokyo, 1994.

[Breukers and Hoekstra 2004] Breukers, J., and Hoekstra, R. J.: *Epistemology and ontology in core ontologies: FOLaw and LRI-Core, two core ontologies for law*. In: Gangemi, A., Borgo, S.: Proceedings of EKAW Workshop on Core ontologies. CEUR, Aachen, 2004.

[Brewer 1996] Brewer, S.: Exemplary reasoning: Semantics, pragmatics, and the rational force of legal argument by analogy. In: Harvard Law Review, 1996, 923-1028.

[Brewka and Gordon 2010] Brewka, G., Gordon, T. F.: *Carneades and abstract dialectical frameworks: A reconstruction*. In: P. Baroni, F. Cerutti, M. Giacomin, and G.R. Simari (eds): Computational Models of Argument: Proceedings of COMMA 2010, IOS Press, Amsterdam, 2010, 3-12.

[Brewka et. al. 2010] Brewka, G., Eiter, T. and Fink, M.: *Nonmonotonic multi-context systems: A flexible approach for integrating heterogeneous knowledge sources*. In: Logic programming, knowledge representation, and nonmonotonic reasoning. Springer, Berlin Heidelberg, 2011, 233-258.

[Bruninghaus and Ashley 2003] Bruninghaus, S., Ashley, K. D.: *Predicting outcomes of case based legal arguments*. In: Proceedings of the 9th international conference on Artificial intelligence and law, ACM, 2003, 233-242.

[Brüninghaus and Ashley 2005] Brüninghaus, S., Ashley, K. D.: *Generating legal arguments and predictions from case texts*. In: ICAIL 2005, ACM Press, New York, 2005, 65-74.

[Caminada 2006] Caminada, M.: *On the issue of reinstatement in argumentation*. In: Logics in artificial intelligence, Springer, Berlin Heidelberg, 2006, 111-123.

[Carenini and Moore 2001] Carenini, G. and Moore, J.: *An Empirical Study of the Influence of User Tailoring on Evaluative Argument Effectiveness*. In: Proceedings of the 17th International Joint Conference on

Artificial Intelligence (IJCAI 2001), 2001.

[Carnap 1947] Carnap, R.: *Meaning and Necessity*. Univeristy of Chicago Press, 1947.

[Casanovas et al. 2005] Casanovas, P., Poblet, M., Casellas, N., Contreras, N., Benjamins, R., Blasquez, M.: *Supporting newly-appointed judges: A legal knowledge management case study*. In: Journal of Knowledge Management, 2005, 7-27.

[Casellas 2011] Casellas, N.: *Legal Ontology Engineering Methodologies, Modelling Trends, and the Ontology of Professional Judicial Knowledge*. In: Law, Governance and Technology Series, Vol. , 3, 2011.

[Ceci 2012] Ceci, M.: *Combining Ontologies and Rules to Model Judicial Interpretation*. In: Proceedings of the RuleML@ECAI 6th International Doctoral Consortium, vol. 874, CEUR, 2012.

[Ceci and Gordon 2012] Ceci, M., Gordon, T. F.: *Browsing case-law: An application of the Carneades Argumentation System*. In: Proceedings of the RuleML2012@ECAI Challenge, vol. 874, CEUR, 2012, 79-95.

[Ceci and Palmirani 2012] Palmirani, M., Ceci, M.: *Ontology Framework for Judgement Modelling*. In: AI Approaches to the Complexity of Legal Systems. Models and Ethical Challenges for Legal Systems, Legal language and Legal Ontologies, Argumentation and Software Agents, LNCS vol. 7639, Springer, Berlin, 2012, 116-130.

[Chesfievar et al. 2000] Chesnevar, C., Maguitman, G., Loui, R.: *Logical Models of Argument*, ACM Computing Surveys 32 (4), 2000, 337-383.

[Chomsky 2005] Chomsky, N.: *Government of the Future*, Seven Stories Press, New York, 2005.

[Chomsky 2006] Chomsky, N.: *Language and mind*. Cambridge University Press, 2006.

[Colmerauer et al. 1972] Colmerauer A., Kanoui, H., Pasero, R., Roussel, P.: *Un système de communication en français*. Rapport préliminaire de fin de contrat IRIA, Groupe Intelligence Artificielle, Faculté des Sciences de Luminy, Université Aix-Marseille II, France, 1972.

[Conklin 2003] Conklin, J.: *Dialog Mapping: Reflections on an Industrial Strength Case Study*. In: Kirschner, P. A., Buckingham Shum, S. J., Carr, C. S. (eds): Visualizing Argumentation, Springer, 2003 117-136.

[Conte et al. 1977] Conte, A. G., Hilpinen, R., von Wright G. H. (eds): *Deontische Logik und Semantik,* Athenaion, Wiesbaden, 1977.

[Cook and Nguyen 2010] Cook, S., Nguyen, P.: *Logical foundations of proof complexity*. Cambridge University Press, 2010.

[Costa et al. 1998] Costa, M., Sousa, O., Neves, J.: *An architecture to legal distributed case representation*. In: Legal Knowledge Based Systems: JURIX: The Eleventh Conference, 1998.

[Couturat 1903] Couturat, L. (ed): Opuscules et fragments inédits de Leibniz. Alcan, Paris, 1903.

[Crosswhite et al. 2003] Crosswhite, J., Fox, J., Reed, C., Scaltsas, T., Stumpf, S.: *Computational Models of Rhetorical Argument*. In: Reed, C., Norman, T. (eds): Argumentation Machines, Kluwer, 2003, 175-210.

[Cuninghame 2009] Cuninghame, K.: *Guidelines for Legislative Libraries.* De Gruyter Saur, Munich, 2009.

[Davis 1958] Davis, M.: *Computability and Unsolvability*. McGraw-Hill Book Company Inc, New York, 1958.

[Decker et al. 2000] Decker, S., Melnik, S., van Harmelen, F., Fensel, D., Klein, M., Broekstra, J., Erdmann, M., Horrocks, I.: *The Semantic Web: The Roles of XML and RDF*. In: IEEE Internet Computing 4, 2000, 63-74.

[Dennett 1989] Dennett, D. C.: *The intentional stance*. MIT press, Cambridge, MA, 1989.

[Despres and Szulman 2004] Despres, S., Szulman, S.: *Construction of a Legal Ontology from a European Community Legislative text*. In: Legal Knowledge and Information Systems. Jurix 2004: The Seventeenth Annual Conference, Amsterdam, 2004.

[Dittrich et al. 1995] Dittrich, K. R., Gatziu, S., Geppert, A.: *The Active Database Management System Manifesto: A Rulebase of ADBMS Features*. In: Lecture Notes in Computer Science 985, Springer, 1995, 3-20.

[Dung 1995] Dung, P.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming, and n-person games. In: Artificial Intelligence, 77, 1995, 321-357.

[Ellson et. al. 2001] Ellson, J., Gasner, E., Koutsofios, L., North, S. C., Woodhull, G.: *Graphviz – open source graph drawing tools*. In:

Proceedings of the 9th international Symposium on Graph Drawing (GD 2001), Vienna, 2001, 483-494.

[Engisch 1960] Engisch, K.: *Logische Studien zur Gesetzesanwendung*. Winter, Heidelberg, 1960.

[Forgy 1982] Forgy, C. L.: Rete: A fast algorithm for the many pattern/many object pattern match problem. In: Artificial Intelligence, Volume 19, 1/1982, 17-37.

[Frege 1970] Frege, G.: *Begriffsschrift: A formula language, modeled upon that of arithmetic, for pure thought*. In: van Heijenoort, J. (ed): Frege and Gödel: Two Fundamental Texts in Mathematical Logic, Harvard University Press, Cambridge, MA, 1970, 1–82.

[Gangemi 2007] Gangemi, A.: *Design Patterns for Legal Ontology Construction*. In: Trends in Legal Knowledge. The Semantic Web and the Regulation of Electronic Social Systems, European Press Academic Publishing, 2007, 171-191.

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

[Gangemi et al. 2001] Gangemi, A., Guarino, N., Masolo, C., Oltramari, A.: *Understanding top-level ontological distinctions*. In: Proceedings of IJCAI-01 Workshop on Ontologies and Information, 2001.

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

[Gardner 1987] Gardner, A.: Artificial Intelligence Approach to Legal Reasoning. MIT Press, Cambridge, MA, 1987.

[Garssen 2001] Garssen, B.: *Argument schemes*. In: van Eemeren, F. H. (ed): Critical Concepts in Argumentation Theory, Amsterdam University Press, 2001, 81-100.

[Gelfond and Lifschitz 1988] Gelfond, M., Lifschitz, V.: *The stable modelsemantics for logic programming*. In: Kowalski, R. A., Bowen, K. A. (eds): Logic Programming. Proceedings of the Fifth International Conference and Symposium. The MIT Press, Cambridge, MA, 1988, 1070-1080

[Giuliani 1961] Giuliani, A.: *Il concetto di prova*. Milano, Giuffrè, 1961.

[Gödel 1929] Gödel, K.: *Über die Vollständigkeit des Logikkalküls*. Doctoral dissertation. University Of Vienna, 1929.

[Goldfarb 1991] Goldfarb, C.: *The SGML Handbook*. Oxford University Press, 1991.

[Goldfarb 1998] Goldfarb, C.: *XML Handbook*. Prentice Hall, Upper Saddle River, NJ,1998

[Gordon 1994] Gordon, T. F.: *Computational dialectics*. In: First Workshop on Computational Dialectics. Twelfth National Conference on Artificial Intelligence-AAAI, vol. 94, 1994.

[Gordon 1995] Gordon, T. F.: The Pleadings Game: an artificial intelligence model of procedural justice. Kluwer Academic Publishers, Dordrecht, 1995.

[Gordon 2005] Gordon, T. F.: *A computational model of argument for legal reasoning support systems*. In: Dunne, P. E., Bench-Capon, T. J. M. (eds): Argumentation in Artificial Intelligence and Law, IAAIL Workshop Series, Wolf Legal Publishers, Nihmegen, The Netherlands, 2005, 53-64.

[Gordon 2006] Gordon, T. F., Walton, D.: *The Carneades Argumentation Framework: using presumptions and exceptions to model critical questions*. In: Proceedings of the First International Conference on Computational models of Argument (COMMA 06), Amsterdam, IOS Press, 2006.

[Gordon 2007a] Gordon, T. F.: Constructing Arguments with a Computational Model of an Argumentation Scheme for legal Rules: Interpreting legal rules as reasoning policies. In: Association for Computing Machinery (ACM): ICAIL 2007, 11th International Conference on Artificial Intelligence and Law. Proceedings: 04-08 June 2007, Standford, CA, ACM Press, New York, 2007, 117-121

[Gordon 2007b] Gordon, T. F.: *Visualizing Carneades argument graphs*. In: Law, Probability and Risk, 6(1-4), 2007, 109–117

[Gordon 2008] Gordon, T. F.: *Constructing Legal Arguments with Rules in the Legal Knowledge Interchange Format (LKIF)*. In: Casanovas, P., Sartor, G., Casellas, N., Rubino, R. (eds): Computable Models of the Law, Languages, Dialogues, Games, Ontologies. Springer, Berlin, 2008, 162-184.

[Gordon 2010] Gordon, T. F.: *An Overview of the Carneades Argumentation Support System*. In: Tindale, C. W., Reed, C.: Dialectics, Dialogue and Argumentation. An Examination of Douglas

Walton's Theories of Reasoning. College Publications, 2010, 145-156.

[Gordon 2011a] Gordon, T. F.: Hybrid reasoning with argumentation schemes, 2011. In: Proceedings of the 8th Workshop on Computational Models of Natural Argument (CMNA 08), 2008, 16-25.

[Gordon 2011b] Gordon, T. F.: *Combining Rules and Ontologies with Carneades*. In: Proceedings of the 5th International RuleML2011@BRF Challenge, CEUR Workshop Proceedings, 2011, 103-110.

[Gordon and Karacapilidis 1997] Gordon, T. F., Karacapilidis, N.: *The Zeno Argumentation Framework*. In: Proceedings of the ICAIL'97 Conference, ACM Press, New York, 1997, 10-18.

[Gordon and Walton 2006] Gordon, T. F., Walton, D.: *Pierson vs. Post revisited - a reconstruction using the Carneades Argumentation Framework*. In: Dunne, P. E., Bench-Capon, T. J. M.: Proceedings of the First International Conference on Computational Models of Argument (COMMA 06), IOS Press, Liverpool, 2006.

[Gordon and Walton 2006] Gordon, T. F., Walton, D.: *The Carneades argumentation framework*. In: Frontiers in Artificial Intelligence and Applications 144, 2006, 195.

[Gordon and Walton 2009] Gordon, T. F., Walton, D.: *Legal Reasoning with Argumentation Schemes*. In: Proceedings of the Twelfth International Conference on Artificial Intelligence and Law, ACM Press, New York, 2009, 137-146.

[Gordon and Walton 2012] Gordon, T. F., Walton, D.: *A Carneades Reconstruction of Popov vs. Hayashi*. In: Artificial Intelligence and Law 20, 1/2012.

[Gordon et al. 2007] Gordon, T. F., Prakken, H., Walton, D.: *The Carneades model of argument and burden of proof*. In: Artificial Intelligence 171, 10/2007, 875-896.

[Gordon et al. 2009] Gordon, T. F., Governatori, G., Rotolo, A.: *Rules and Norms: Requirements for Rule Interchange Languages in the Legal Domain*. In: Rule Interchange and Applications, International Symposium, RuleML 2009, Springer, Berlin, 2009, 282-296.

[Governatori 2011] Governatori, G.: *On the relationship between Carneades and Defeasible logic*. In: Proceedings of the 13th International Conference on Artificial Intelligence and Law (ICAIL 2011), ACM, New York, 2011.

[Governatori and Rotolo 2004] Governatori, G., Rotolo, A.: *Defeasible Logic: Agency, Intention and Obligation*. In: Deontic Logic in Computer Science, Springer, Berlin, 2004.

[Governatori and Rotolo 2008] Governatori, G., Rotolo, A.: *BIO logical agents: Norms, beliefs, intentions in defeasible logic*. In: Autonomous Agents and Multi-Agent Systems, 17(1), 2008, 36-69.

[Governatori et al. 2005] Governatori, G., Rotolo, A., Sartor, G.: *Temporalised normative positions in defeasible logic*. In: Proceedings of the 10th international conference on Artificial intelligence and law, ACM, 2005, 25-34.

[Grasso et al. 2000] Grasso, F., Cawsey, A., Jones, R.: *Dialectical Argumentation to Solve Conflicts in Advice Giving: A Case Study in the Promotion of Healthy Nutrition*. In: International Journal of Human-Computer Studies, 53 (6), 2000, 1077-1115.

[Gruninger and Fox 1994] Gruninger, M., Fox, M.: *The role of competency questions in enterprise engineering*. In: Proceedings of the IFIP WG5.7 Workshop on Benchmarking Theory and Practice, Trondheim, Norway, 1994.

[Guarino and Welty 2002] Guarino, N., Welty, C.: *Evaluating Ontological Decisions with OntoClean*. In: Communications of the ACM, 45(2), 2002, 61-65.

[Haft 1990] Haft, F.: *Juristiche Rhetorik*. Alber, Freiburg, 1990.

[Hage 1997] Hage, J. C.: Reasoning with Rules – An Essay on Legal Reasoning and its Underlying Logic. Kluwer Academic Publishers, Dordrecht, 1997.

[Hart 1961] Hart, H. L. A.: *The Concept of Law*. Clarendon Law Series, Oxford, 1961.

[Hart 1970] Hart, H. L. A. (ed), Bentham, J.: *Of Laws in General*. The Athlone Press, London, 1970.

[Hart 1982] Hart, H. L. A.: *Essays on Bentham*. Clarendon, Oxford, 1982.

[Hart 1983] Hart, H. L. A.: *Essays in Jurisprudence and Philosophy*. Clarendon Press, Oxford, 1983.

[Hastings 1963] Hastings, A. C.: *A Reformulation of the Modes of Reasoning in Argumentation*. PhD thesis, Northwestern University, Evanston, Illinois, 1963.

[Hegel 1820] Hegel, G. W. F.: *Grundlinien der Philosophie des Rechts*. Felix Meiner Verlag, Hamburg, 2013 (1st ed. 1820). Available at http://archive.org/details/grundlinienderp00gansgoog.

[Henderson and Bench-Capon 2001] Henderson, J., Bench-Capon, T. J. M.: *Dynamic arguments in a case law domain*. In: Proceedings of the 8th international conference on Artificial intelligence and law, ACM, 2001, 60-69.

[Hermann 1997] Herrmann, C. S.: *Symbolical reasoning about numerical data: A hybrid approach.* In: Applied Intelligence 7, 4/1997, 339-354.

[Hilbert and Ackermann 1927] Hilbert, D., Ackermann, W.: *Grundzüge der theoretischen Logik*. Springer, Berlin, 1927.

[Hintikka 1957] Hintikka, J.: *Quantifiers in Deontic Logic*. In: Societas Scientiarum Fennica, Commentationes humanarum litterarum, 23, 1957.

[Hirsch et al. 2004] Hirsch, L., Saeedi, M., Cornillon, J., Litosseliti, L.: *A Structured Dialogue Tool for Argumentative Learning*. In: Journal of Computer Assisted Learning 20, 2004, 72-80.

[Hoekstra et al. 2007] Hoekstra, R., Breuker, J., Di Bello, M., Boer, A.: *The LKIF Core Ontology of Basic Legal Concepts*. In: Casanovas, P., Biasiotti, M. A., Francesconi, E., Sagri, M. T. (eds): Proceedings of LOAIT 2007.

[Hohfeld and Cook 1919] Hohfeld, W. N., Cook, W. W. (eds): Fundamental Legal Conceptions as Applied in Judicial Reasoning and Other Legal Essays. Yale University Press, 1919.

[Horn 1951] Horn, A.: *On sentences which are true of direct unions of algebras*. In: Journal of Symbolic Logic, 16, 1951, 14–21.

[Horrocks 2003] Horrocks, I.: *Implementation and Optimization Techniques*. In: Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P. F. (eds): The Description Logic Handbook: Theory, Implementation and Applications, Cambridge University Press, 2003.

[IMPACT 2013] IMPACT (FP7-ICT-2009-4; No. 247228) – D4.2 – *Implementation and Documentation of the Policy Modelling Tool*. Available at:

http://ec.europa.eu/information_society/apps/projects/logos/8/247228/080/deliverables/001_IMPACTD42.pdf.

[Kafka 1925] Kafka, F.: *Der Process*. Fischer Verlag, Frankfurt am Main, 2006 (1st ed. 1925). Available Online at http://www.gutenberg.org/ebooks/7849.

[Kanger 1957] Kanger, S.: *Provability in Logic*. Ph.D dissertation, University of Stockholm, 1957.

[Kelsen 1945] Kelsen, H.: *General Theory of Law and the State*. The Lawbook Exchange Ltd., Clark, NJ, 1945.

[Kelsen 1957] Kelsen, H.: *What is Justice? Justice, Law and Politics in the Mirror of Science*. University of California Press, Berkeley, 1957.

[Kirschner et al. 2003] Kirschner, P. A., Buckingham Shum, S. J., Carr, C. S. (eds): *Visualizing Argumentation*. Springer, Berlin, 2003.

[Knuth 2011] Knuth, D. E.: *Leaders in computing: Changing the digital world*. BCS, The Chartered Institute, 2011.

[Kontopoulos et al. 2011] Kontopoulos, E., Bassiliades, N., Governatori, G., Antoniou, G.: *A Modal defeasible Reasoner of Deontic Logic for the Semantic Web*. In: International Journal on Semantc Web and Information Systems, 2011, 18-43.

[Kripke 1959] Kripke, S.: *A Completeness Theorem in Modal Logic*. In: Journal of Symbolic Logic, 24, 1959.

[Kunz and Rittel 1970] Kunz, W., Rittel, H. W. J.: *Issues as elements of information systems*. Working Paper 131, Institute for Urban and Regional Development, University of California, Berkeley, 1970. Available at http://iurd.berkeley.edu/sites/default/files/wp/131.pdf .

[Lam and Governatori 2009] Lam, H.-P., Governatori, G.: *The making of SPINdle*. In: Rule Interchange and Applications, Springer, Berlin Heidelberg, 2009, 315-322.

[Lam and Governatori 2010] Lam, H.-P., Governatori, G.: *On the problem of computing ambiguity propagation and well-founded semantics in defeasible logic*. In: Rotolo, A., Hall, J., Dean, M., Tabet, S. (eds): Proceedings of the 4th International Web Rule Symposium: Research Based and Industry Focued (RuleML-2010). Washington, DC, 2010.

[Lee and Sohn 2003] Lee, J. K., Sohn, M. M.: *The eXtensible Rule Markup Language*. In: Communications of the ACM, Volume 46, Issue 5, 2003, 59-64.

[Lehmann 2003] Lehmann, J.: Causation in Artificial Intelligence and Law: a modeling approach. PhD thesis, University of Amsterdam, 2003.

[Leibniz 1666] Leibniz, G. W.: *Dissertatio de arte combinatoria*, 1666. In: Sämtliche Schriften und Briefe. Akademie Verlag, Berlin, 1694, A VI 1, p.163

[Leibniz 1694] Leibniz, G. W.: *Sämtliche Schriften und Briefe*. Akademie Verlag, Berlin, 2004 (1st ed. 1694).

[Leibniz 1974] Leibniz, G. W.: Nova methodus discendae docendaeque jurisprudentiae, ex artis didacticae principiis in parte generali praemissis, experientiae luce. Detlev Auvermann KG, Glasütten im Taunus, 1974 (1st ed. 1748).

[Lemmon and Scott 1977] Lemmon, D. S., Scott, E. J.: *An Introduction to Modal Logic*. In: Scott, E. J., Lemmon, D. S.: An introduction to modal logic: The Lemmon Notes. Blackwell, 1977.

[Levesque and Brachman 1987] Levesque, H. J., Brachman, R. J.: *Expressiveness and tractability in knowledge representation and reasoning*. In: Computational Intelligence, 3, 1987, 78-93.

[Levi 1948] Levi, E. H.: *An Introduction to Legal Reasoning.* University of Chicago Press, 1948.

[Lewis 1918] Lewis, C. I.: *A Survey of Symbolic Logic*. University of California Press, 1918.

[Lima et al. 2008] Lima, J. A. O., Palmirani, M., Vitali, F.: *A Time-aware Ontology for Legal Resources***.** In: Proceedings of the Tenth International ISKO Conference, 2008.

[Losano 2000] Losano, M. G.: I grandi sistemi giuridici. Introduzione ai diritti europei ed extraeuropei. Laterza, Roma-Bari, 2000.

[Macagno et al. 2007] Macagno, F., Reed, C., Walton, D.: *Argument Diagramming in Logic, Artificial Intelligence, and Law*. In: Artificial Intelligence and Law, 2007, 87-109.

[Maher 2001] Maher, M. J.: *Propositional defeasible logic has linear complexity*. In: Theory and Practice of Logic Programming 1(6), 2001, 691–711.

[Masolo et al. 2002] Masolo, C., Borgo, S., Gangemi, A., Guarino, N., Oltramari, A., Oltramari, R., Schneider, L., Horrocks, I.: *Wonderweb deliverable d17. the wonderweb library of foundational ontologies and the dolce ontology*. (2002).

[McCarthy 1986] McCarthy, J.: *Applications of circumscription to common sense reasoning*. In: Artificial Intelligence 28(1), 1986, 89-116.

[McCarthy 1995] McCarty, L.: *An implementation of Eisner v. Macomber*. In: Proceedings of the Fifth International Conference on Artificial Intelligence and Law, ACM Press, New York, 1995, 276-286.

[McCormack 2005] McCormack, W.: *Lochner, Liberty, Property, and Human Rights*. In: NYU Journal of Law & Liberty 1, 2005, 433.

[Minh et al. 2009] Minh, D. T., Eiter, T., Krennwallner, T.: *Realizing Default Logic over Description Logic Knowledge Bases*. In: Proceeding of the ECSQARU 2009.

[Mommers 2010] Mommers, L.: *Ontologies in the Legal Domain*. In: Poli, R., Seibt, J. (eds): Theory and Applications of Ontology: Philosophical Perspectives, Springer, Berlin, 2010, 265-276.

[Motik 2006] Motik, B.: Reasoning in description logics using resolution and deductive databases, PhD thesis, Universität Karlsruhe, 2006.

[Nebel 1988] Nebel, B.: *Computational complexity of terminological reasoning in BACK*. In: Artificial Intelligence, 32(3), 1988, 371-383.

[Nejdl et al. 2000] Nejdl, W., Wolpers, M., Capelle, C.: *The RDF Schema Specification revisited*. In: Workshop Modellierung 2000, St. Goar, Germany, 2000.

[Niles and Pease 2001] Niles, I., Pease, A.: *Towards a standard upper ontology*. In: Proceedings of the international conference on Formal Ontology in Information Systems - Volume 2001 (FOIS '01), ACM, New York, NY, 2001, 2-9.

[Novák 1999] Novák, V., Perfilieva, I., Močkoř, J.: *Mathematical principles of fuzzy logic*. Kluwer Academic, Dodrecht, 1999.

[Nute 1994] Nute, D. N.: *Defeasible Logic*. In: Gabbay, D. M., Hogger, C. J., Robinson, J. A. (eds): Handbook of Logic in Artificial Intelligence and Logic Programming, Vol. 3: Nonmonotonic Reasoning and Uncertain Reasoning, Oxford University Press, 1994, 355-395.

[Nute 1998] Nute, D. N.: *Norms, Priorities, and Defeasible Logic*. In: Norms, Logics and Information Systems, IOS Press, 1998, 201-218.

[Okada et al. 2008] Okada, A., Shum, S. J. B. and Sherborne, T. (eds): *Knowledge Cartography: software tools and mapping techniques*. Springer, Berlin, 2008.

[Palmer 2009] Palmer, S. B.: *The Early History of HTML*. Available at: http://infomesh.net/html/history/early/. Retrieved 2013-04-30.

[Palmirani 2005] Palmirani, M.: Dynamics of Norms over Time: a Model for Legislative Consolidation. In: I Quaderni, 18, 2005, 42-69.

[Palmirani and Benigni 2007] Palmirani, M., Benigni, F.: *Norma-system: A legal information system for managing time*. In: Proceedings of the V Legislative XML Workshop, 2007, 205-224.

[Palmirani and Brighi 2010] Palmirani, M., Brighi, R.: *Model Regularity of Legal Language in Active Modifications*. In: Lecture Notes in Computer Science, Springer, Berlin, 2010, 54-73

[Palmirani et al. 2009] Palmirani, M., Contissa, G., Rubino, R.: *Fill the Gap in the legal Knowledge Modelling*. In: Proceedings of RuleML 2009, LNCS 5858, Springer, Berlin, 2009, 305-314.

[Palmirani et al. 2010] Palmirani, M., Governatori, G., Contissa, G.: *Temporal Dimensions in Rules Modelling*. In: Proceedings of the 2010 conference on Legal Knowledge and Information Systems: JURIX, 2010, 159-162

[Palmirani et al. 2011a] Palmirani, M., Governatori, G., Contissa, G.: *Modelling temporal legal rules*. In: Proceedings of the 13th International Conference on Artificial Intelligence and Law, ACM, 2011, 131-135.

[Palmirani et al. 2011b] Palmirani, M., Ceci, M., Radicioni, D., Mazzei, A.: *FrameNet model of the suspension of norms*. In: The 13th International Conference on Artificial Intelligence and Law, Proceedings of the Conference, ACM, Pittsburgh, 2011, 179-193.

[Palmirani et al. 2011c] Palmirani, M., Governatori, G., Rotolo, A., Tabet, S., Boley H., Paschke, A.: *LegalRuleML: XML-Based Rules and Norms*. In: RuleML 2011, 298-312.

[Palmirani et al. 2012] Palmirani, M., Ognibene, T., Cervone, L.: *Legal rules, text, and ontologies over time.* In: Proceedings of the RuleML@ ECAI 6th International Rule Challenge, 2012.

[Paolucci et al. 1995] Paolucci, M., Suthers, D., Weiner, A.: *Belvedere: Stimulating Students' Critical Discussion*. In: Proceedings of CHI'95, ACM Press, 1995, 123-4.

[Park et al. 1998] Park, R. C., Leonard, D. P., Goldberg, S. H.: *Evidence Law*. West Group, St. Paul, Minnesota, 1998.

[Pashke 2007] Paschke, A.: Rule-Based Service Level Agreements - Knowledge Representation for Automated e-Contract, SLA and Policy Management. IDEA, Munich, 2007.

[Pashke et al. 2006] Paschke, A., Kozlenkov, A., Boley, H., Kifer, M., Tabet, S., Dean, M., Barrett, K.: *Reaction ruleml*. Available at: http://ibis.in.tum.de/research/reactionruleml/.

[Pashke et al. 2007] Paschke, A., Kozlenkov, A., Boley, H.: *A homogeneous reaction rule language for complex event processing*. In: Proceedings of the 2nd International Workshop on Event Drive Architecture and Event Processing Systems (EDA-PS 2007).

[Patel-Schneider 1989] Patel-Schneider, P. F.: *Undecidability of subsumption in NIKL*. In: Artificial Intelligence, 39, 1989, 263-272.

[Pattaro et al. 2005] Pattaro, E., Rottleuthner, H., Shiner, R. A., Peczenik, A., Sartor, G. and Rotolo, A.: *A Treatise of legal philosophy and general jurisprudence*. Vol. 18. Springer, Berlin, 2005.

[Paulson 1990] Paulson, S. L.: *Kelsen on legal interpretation*. In: Legal Studies, 10, 1990, 136–152.

[Pearl 1984] Pearl, J.: Heuristics: Intelligent Search Strategies for Computer Problem Solving. Addison-Wesley, Reading, MA, 1984.

[Perelman 1979] Perelman, C.: Logique juridique: Nouvelle rhétorique. Dalloz, Paris, 1979.

[Perelman and Olbrechts-Tyteca 1969] Perelman, C., Olbrechts-Tyteca, L.: *The New Rhetoric: A Treaty on Argumentation*. University of Notre Dame Press, Notre Dame, IN, 1969 (1st ed. in French 1958).

[Pollock 1995] Pollock, J.: *Cognitive Carpentry*. MIT Press, Cambridge, 1995.

[Pollock 2002] Pollock, J.: Defeasible Reasoning with Variable Degrees of Justification II, greatly revised and expanded version of ‘Defeasible Reasoning with Variable Degrees of Justification’. In: Artificial Intelligence 133, 2002 233-282. Available at: http://oscarhome.soc-sci.arizona.edu/ftp/PAPERS/Degrees.pdf

[Poser 1969] Poser, H.: Zur Theorie der Modalbegriffe bei G. W. Leibniz. Steiner, Wiesbaden, 1969.

[Prakken 1995] Prakken, H.: *From logic to Dialectic in Legal Argument*. In: Proceedings of the 5th international conference on Artificial intelligence and law, ACM, 1995, 165-174.

[Prakken 1997] Prakken, H.: Logical Tools for Modelling Legal Argument. A Study of Defeasible Reasoning in Law. Kluwer Law and

Philosophy Library Dordrecht, 1997.

[Prakken 2001] Prakken, H.: Relating protocols for dynamic dispute with logics for defeasible argumentation. In: Synthese 127, n. 1-2/2001, 187-219.

[Prakken 2008] Prakken, H.: *Formalizing Ordinary legal Disputes: a Case Study*. In: Artificial Intelligence and Law, 2008, 333-359.

[Prakken 2008] Prakken, H.: *Combining modes of reasoning: An application of abstract argumentation*. In: Proceedings of the 11th european conference on Logics in Arrtificial Intelligence (JELIA 2008), volume 5293 of Springer Lecture Notes in AI, Springer, Berlin, 2008, 349-361.

[Prakken 2010] Prakken, H.: An Abstract Framework for Argumentation with Structured Arguments. In: Argument and Computation, 2010, 93-124.

[Prakken 2012] Prakken, H.: Reconstructing Popov vs. Hayashi in a framework for argumentation with structured arguments and Dungean semantics. In: Artificial Intelligence and Law 20(1), 2012, 57-82.

[Prakken and Sartor 1996] Prakken, H., Sartor, G.: *A dialectical model of assessing conflicting arguments in legal reasoning.* In: Artificial Intelligence and law, 4, 1996, 331-368.

[Prakken and Sartor 1996] Prakken, H., Sartor, G.: *A dialectical model of assessing conflicting argument in legal reasoning*. In: Artificial Intelligence and Law, 4(3-4), 1996, 331-368.

[Prakken and Sartor 1997] Prakken, H., Sartor, G.: *Argument-based extended logic programming with defeasible priorities*. In: Journal of Applied Non-Classical Logics, 7, 1997, 25-75.

[Prakken and Sartor 2006] Prakken, H., Sartor, G.: Presumptions and burden of proof. In: van Engers, T. (ed): Legal Knowledge and Information Systems, JURIX 2006: The Nineteenth Annual Conference, IOS Press, Amsterdam, 2006, 21-30.

[Prakken and Sartor, 2009] Prakken, H., Sartor, G.: *A Logical Analysis of Burdens of Proof*. In: Kaptein, H., Prakken H., Verheij, B. (eds): Legal Evidence and Proof: Statistics, Stories, Logic, Applied Legal Philosophy Series, Ashgate Publishing, Farnham, 2009, 223-253.

[Rahwan and Larson 2009] Rahwan, I., Larson, K.: *Argumentation and game theory*. In: Argumentation in Artificial Intelligence, Springer US, 2009, 321-339.

[Rapisarda and Willems 2006] Rapisarda, P., Willems, J. C.: *Recent Developments in Behavioral System Theory*. In: Proceedings of MTNS, July 24-28, Kyoto, 2006.

[Reed and Rowe 2004] Reed, C., Rowe, G.: *Araucaria: Software for Argument Analysis, Diagramming and Representation*. In: International Journal of AI Tools 14 (3-4), 2004, 961-980.

[Reed and Walton 2007] Reed, C., Walton, D.: *Argumentation schemes in dialogue*. In: Dissensus and the Search for Common Ground (Proceedings of OSSA 2007), 2007.

[Reed et al. 2007] Reed, D. Walton, Macagno F.: *Argument diagramming in logic, law and artificial intelligence*. In: Knowledge Engineering Review, 22(1), 2007, 87–109.

[Reiter 1978] Reiter, R.: *On Closed World Databases*. In: Gallaire, H., Minker, J. *Logic and Data Bases*, Plenum Press, 1978, 119-140.

[Reiter 1980] Reiter, R.: *A logic for default reasoning*. In: Artificial Intelligence, 13, 1980, 81-132.

[Ren et al. 2010] Ren, Y., Pan, J. Z., Zhao, Y.: *Closed World Reasoning for OWL2 with NBox*. Tsinghua Science & Technology, 2010, 692-701.

[Rissland and Ashley 1987] Rissland, E. L., Ashley, K. D.: *A case-based system for trade secrets law*. In: Proceedings of the 1st international conference on Artificial intelligence and law, ACM, 1987, 60-66.

[Rissland et al. 2003] Rissland, E. L., Ashley, K. D., Branting, L. K.: *Case-based reasoning and law*. In: Knowledge Engineering Review 20, no. 3, 2005, 293-298.

[Rittel 1972] Rittel, H. W. J.: *On the planning crisis: Systems analysis of the "first and second generations"*. In: Bedrifts Okonomen, 8, 1972, 390–396. Also available as Reprint 107, Institute of Urban and Regional Development, University of California, Berkeley, at: http://iurd.berkeley.edu/sites/default/files/pubs/RP107.pdf.

[Rittel and Noble 1989] Rittel, H. W. J., Noble, D.: *Issue-based information systems for design*. Working Paper 492, Institute of Urban and Regional Development, University of California, Berkeley, 1989, available at:

http://iurd.berkeley.edu/sites/default/files/wp/492.pdf.

[Rowe et al. 2006] Rowe, G., Macagno, F., Reed, C., Walton, D.: *Araucaria as a Tool for Diagramming Arguments in Teaching and Studying Philosophy*. In: Teaching Philosophy 29:2, 2006, 111-124.

[Russell 1903] Russell, B.: *The Principles of Mathematics*. Cambridge University Press, 1903.

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

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

[Sartor 2005] Sartor, G.: Legal Reasoning – A Cognitive Approach to the Law. In [Pattaro et al. 2005].

[Sartor 2008] Sartor, G.: *Legal Validity: An Inferential Analysis*. In: Ratio Juris 21, 2008, 212–247.

[Sartor 2009] Sartor, G.: *Legal Concepts as Inferential Nodes and Ontological Categories*. In: Artificial Intelligence and Law, 2009, 217-251.

[Sartor 2012] Sartor, G.: *Defeasibility in Legal Reasoning*. In: Ferrer Beltran, J., Ratti, G. B.: The Logic of Legal Requirements: Essays on Defeasibility, Oxford University Press, Oxford, 2012.

[Sartor et al. 2011] Sartor, G., Palmirani, M., Francesconi, E., Biasiotti, M. A. (eds): *Legislative XML for the Semantic Web: Principles, Models, Standards for Document Management*. In: Law, Governance and Technology Series, Vol. 4, Springer 2011.

[Savvas and Bassiliades 2009] Savvas, I., Bassiliades, N.: A Process-Oriented Ontology-Based Knowledge Management System for Facilitating Operational Procedures in Public Administration. In: Expert Systems with Applications, 2009, 4467-4478.

[Schmidt-Schauß 1989] Schmidt-Schauß, M.: *Subsumption in KL-ONE is undecidable*. In: Brachman, R. J., Levesque, H. J., Reiter, R. (eds): Proceedings Of the 1st International Conference on the Principles of Knowledge Representation and Reasoning (KR 1989), Morgan Kaufmann, Los Altos, 1989, 421-431.

[Schmidt-Schauß and Smolka Schmidt-Schauß, M., Smolka, G.: *Attributive concept*

1991] *descriptions with complements*. In: Artificial Intelligence, 48(1), 1991, 1-26.

[Schreiber et al. 2000] Schreiber, G., Akkermans, H., Anjewierden, A., de Hoog, R., Shadbolt, N., Van den Velde, W., Wielinga, B.: *Knowledge Engineering and Managament: the CommonKADS Methodology*. MIT Press, Cambridge, MA, 2000.

[Schum 1994] Schum, D.: Evidential Foundations of Probabilistic Reasoning. John Wiley & Sons, Toronto, 1994.

[Schum et al. 2005] Schum, D., Anderson, T., Twining, W.: *Analysis of Evidence*. Cambridge University Press. 2005.

[Scriven 1976] Scriven, M.: *Reasoning*. Edgepress, Point Reyes, 1976.

[Searle 1969] Searle, J. R.: *Speech Acts: an Essay in the Philosophy of Language*. Cambridge University Press, 1969.

[Searle 1975] Searle, J. R.: *A Taxonomy of Illocutionary Acts*. In: Gunderson, K. (ed): Language, Mind, and Knowledge. University of Minnesota Press, Minneapolis, 1975, 344–369.

[Searle 1980] Searle, J. R.: *Minds, brains, and programs*. In: Behavioral and brain sciences 3, no. 3, 1980, 417-457.

[Searle 1983] Searle, J. R.: *Intentionality*. Cambridge University Press, Cambridge, 1983.

[Selvin 2003] Selvin, A. M.: *Fostering Collective Intelligence: Helping Groups Use Visualized Argumentation*. In: Kirschner, P. A., Buckingham Shum, S. J., Carr, C. S. (eds): Visualizing Argumentation. Springer, Berlin, 2003, 137-163.

[Sen 1999] Sen, A.: *Development as Freedom*. Random House, New York, 1999.

[Shaheed et al. 2005] Shaheed, J. Yip, A., Cunningham, J.: *A Top -Level Language-Biased Legal Ontology*. In: LOAIT – Legal Ontologies and Artificial Intelligence Techniques, Bologna, 2005.

[Shakespeare 1603] Shakespeare, W.: *Hamlet*. In: Thompson, A., Taylor, N.: Hamlet: The Texts of 1603 and 1623: Third Series. Arden Shakespeare, 2007 (1st edition 1603).

[Sirin and Parsia 2007] Sirin, E., Parsia, B.: *SPARQL-DL: SPARQL Query for OWL-DL*. In: 3rd OWL Experiences and Directions Workshop, 2007.

[Smiraglia 2002] Smiraglia, R. P.: *Further reflections on the nature of 'A Work': An introduction*. In: Cataloging & classification quarterly 33.3-

4, 2002, 1-11.

[Sottara et al. 2010] Sottara, D., Mello, P., Proctor, M.: *A configurable Rete-OO engine for reasoning with different types of imperfect information*. In: Knowledge and Data Engineering, IEEE Transactions on, 22(11), 2010, 1535-1548.

[Spaak 2012] Spaak, T.: *Naturalism and Non-Naturalism in Legal Philosophy: Hägerström on Kelsen*. In: Eliaeson, S., Mindus, P., Turner, S. (eds): Axel Hägerström and modern social thought. The Bardwell Press, Oxford, 2012.

[Stein 1992] Stein, L. A.: *Resolving ambiguity in nonmonotonic inheritance hierarchies*. In: Artificial Intelligence 55(2-3), 1992, 259-310.

[Strong 1992] Strong, J. W.: *McCormick on Evidence*. West Publishing Co., Eagan, 1992.

[Tammelo 1978] Tammelo, I.: *Modem Logic in the Service of Law*. Springer, Vienna, 1978.

[Tarello 1980] Tarello, G.: *L'interpretazione della legge*. Giuffrè, Milano, 1980.

[Tarski 1941] Tarski, A.: *On the calculus of relations*. In: Journal of Symbolic Logic, 6, 1941, 73-89.

[Taruffo 1998] Taruffo, M.: *Judicial decisions and artificial intelligence*. In: Artificial Intelligence and law, 6, 2-4, 1998, 311-324.

[Tiscornia et al. 2011] Tiscornia, D., Agnoloni, T., Sagri, M.-T.: *Balancing rights and values in the Italian Courts: a statistical and conceptual analysis*. In: Law, Probability and Risk 10, no. 3, 2011, 265-275.

[Toulmin 1958] Toulmin, S.: *The Uses of the Argument*. Cambridge University Press, Cambridge, 1958.

[Valente 1995] Valente, A.: *Legal Knowledge Engineering: A Modelling Approach.* University of Amsterdam, IOS Press, Amsterdam, 1995.

[Valente et al. 1999] Valente, A., Breuker, J. A., Brouwer, P. W.: *Legal modelling and automated reasoning with ON-LINE*. In: International Journal of Human Computer Studies, 51, 1999, 1079–1126.

[van Emden and Kowalski 1976] van Emden, M., Kowalski, R.: *The semantics of predicate logic as a programming language*. In: Journal of ACM, Vol. 23, 1976, 733-742.

[van Gelder and Rizzo 2001] van Gelder, T., Rizzo, A.: *Reason!Able Across the Curriculum*. In: 2001: Is IT an Odyssey in Learning? Proceedings of the 2001 Conference of ICT in Education, Victoria, 2001.

[van Gelder 2001] van Gelder, T.: *The Reason! Project*. In: The Skeptic 21 (2), 2001, 9-12.

[van Gelder et al. 1991] van Gelder, T., Ross, K. A., Schlipf, J. S.: *The well-founded semantics for general logic programs*. In: Journal of ACM, vol. 38(3), 1991, 619–649.

[van Gijzel and Prakken 2001] van Gijzel, B., Prakken, H.: *Relating Carneades with abstract argumentation*. In: Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Volume Two. AAAI Press, 2011.

[Venn 1880] Venn, J.: *On the Diagrammatic and Mechanical Representation of Propositions and Reasonings*. In: Dublin Philosophical Magazine and Journal of Science 9 (59), 1880, 1-18.

[Verheij 2005] Verheij, B.: Virtual Arguments: On the Design of Argument Assistants for Lawyers and Other Arguers. T.M.C. Asser Press, The Hague, 2005.

[Verheij 1996] Verlheij, B.: Rules, reasons, Arguments. Formal Studies of Argumentation and Defeat. Ph.D. thesis, Universiteit Maastricht, 1996.

[Verheij 2002] Verheij, B.: On the existence and multiplicity of extension in dialectical argumentation. In: Proceedings of the 9th NMR, 2002, 416-425.

[Verheij 2003] Verheij, B.: Deflog: on the logical interpretation of prima facie justified assumptions. In: Journal of Logic and Computation, 2003, 319-346

[Viehweg 1965] Viehweg, T.: Topik und Jurisprudenz. Ein Beitrag zur rechtswissenschaflichen Grundlagenforschung (3 ed.). Beck, Munich, 1965.

[Vitali 1997] Vitali, F.: *Akoma Ntoso Release Notes*. 1997. Available at: http://www.akomantoso.org.

[von Wright 1951] von Wright, G. H.: *Deontic Logic*. In: Mind, New Series, Vol. 60, No. 237, 1951, 1-15.

[W3C Consortium 2012] W3C Consortium, *OWL 2 Web Ontology Language Overview*, 11 December 2012. Available at: www.w3.org/TR/2012/REC-owl2-overview-20121211/.

[Wahlforss and Ljung 2008] Wahlforss, E., Ljung, A.: *People, Profiles & Trust: On interpersonal trust in web-mediated social spaces*. Alexander ljung & Eric Wahlforss, Berlin, 2008. Available at: http://trustmojo.com/media/People-Profiles-And-Trust.pdf.

[Walton 1996] Walton, D.: Argumentation Schemes for Presumptive Reasoning. Erlabum, Mahwah, NJ, 1996.

[Walton 1996b] Walton, D.: Argument Structure: A Pragmatic Theory, Toronto Studies in Philosophy. University of Toronto Press, Toronto, 1996.

[Walton 1997] Walton, D.: *Appeal to Expert Opinion*. Penn State Press, university Park, 1997.

[Walton 1998] Walton, D.: *The New Dialectic*: *Conversational Contexts of Argument*. University of Toronto Press, Toronto/Buffalo, 1998.

[Walton 2001] Walton, D.: *Enthymemes, common knowledge, and plausible inference*. In: Philosophy and Rhetoric 34, 2/ 2001, 93-112.

[Walton 2005] Walton, D., Godden, D.: *The nature of critical questions in argumentation schemes*. In: Hitchcock, D. (ed): The Uses of Argument, McMaster University, Hamilton, ON, Canada, 2005, 476-484.

[Walton 2006] Walton, D.: *Fundamentals of Critical Argumentation*. Cambridge University Press, Cambridge, 2006.

[Walton et al. 2008] Walton, D., Reed, C., Macagno, F.: *Argumentation Schemes*. Cambridge University Press, Cambridge, 2008.

[Weihs 1998] Weihs, J.: *What is a Work?* In: Weihs, J. (ed): The Principles and Future of AACR: Proceedings of the International Conference on the Principles and Future Development of AACR, Toronto, Ontario, Canada, October 23-25, 1997, American Library Association, Chicago, 1998, 62-104.

[Whately 1836] Whately, R.: *Charges and Other Tracts*. Kessinger Publishing, Whitefish, MO, 2008 (1st ed. 1836).

[Wigmore 1931] Wigmore, J. H., *The Principles of Judicial Proof*. Little, Brown and Company, Boston, 2nd edition, 1931.

[Wigmore 1940] Wigmore, J. H., *A Treatise on the Anglo-American System of Evidence*, vol. 1, Little, Brown and Company, Boston, 1940.

[Winkels et al. 2002] Winkels, R., Boer, A., Hoekstra, R.: *CLIME: Lessons Learned in Legal Information Serving*. In: van Harmelen, F. (ed): Proceedings of ECAI-2002. IOS Press, Amsterdam, 2002, 230-

234.

[Wittgenstein 1922] Wittgenstein, L.: Tractatus Logico-Philosophicus. Routledge, London, 1922.

[Wyner and Bench-Capon 2007] Wyner, A., Bench-Capon, T. J. M., Atkison, K. M.: Towards Formalising Argumentation about Legal Cases. In: ICAIL2011, Pittsburg, 5-10 June, ACM, 2011.

[Wyner and Hoekstra 2010] Wyner, A., Hoekstra, R.: *A Legal Case OWL Ontology with an Instantiation of Popov v. Hayashi*. In: The Knowledge Engineering Review, 2010, 1-24.

[Wyner et al. 2007] Wyner, A., Bench-Capon, T. J. M., Atkinson, K.: *Arguments, Values and Baseballs: Representation of Popov v. Hayashi*. In: Proceedings of the 2007 conference on Legal Knowledge and Information Systems (JURIX 2007), Amsterdam, IOS Press, 2007, 151-160.

Appendix A

The Judicial Framework

a.1. The ontology

a.1.1. Core ontology


```
<?xml version="1.0"?>
<!DOCTYPE Ontology [
    <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
    <!ENTITY xml "http://www.w3.org/XML/1998/namespace" >
    <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
    <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
]>
<Ontology xmlns="http://www.w3.org/2002/07/owl#"
     xml:base="http://www.ontologydesignpatterns.org/Judging_Contracts_Core.owl"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
     xmlns:xml="http://www.w3.org/XML/1998/namespace"
     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     ontologyIRI="http://www.ontologydesignpatterns.org/Judging_Contracts_Core.owl">
    <Prefix name="rdf" IRI="http://www.w3.org/1999/02/22-rdf-syntax-ns#"/>
```

```
    <Prefix name="norm" IRI="http://www.estrellaproject.org/lkif-core/norm.owl#"/>

    <Prefix name="owl2xml" IRI="http://www.w3.org/2006/12/owl2-xml#"/>

    <Prefix name="xsd" IRI="http://www.w3.org/2001/XMLSchema#"/>

    <Prefix   name="legal-action"   IRI="http://www.estrellaproject.org/lkif-core/legal-
action.owl#"/>

    <Prefix         name="expression"         IRI="http://www.estrellaproject.org/lkif-
core/expression.owl#"/>

    <Prefix                                                    name="Judging_Contracts_Core"
IRI="http://www.ontologydesignpatterns.org/Judging_Contracts_Core.owl#"/>

    <Prefix name="rdfs" IRI="http://www.w3.org/2000/01/rdf-schema#"/>

    <Prefix name="owl" IRI="http://www.w3.org/2002/07/owl#"/>

    <Import>http://www.estrellaproject.org/lkif-core/lkif-core.owl</Import>

    <Declaration>            <Class  abbreviatedIRI="expression:Communicated_Attitude"/>
</Declaration>

    <Declaration>                     <Class   abbreviatedIRI="expression:Declaration"/>
</Declaration>

    <Declaration>                      <Class   abbreviatedIRI="expression:Expression"/>
</Declaration>

    <Declaration>                   <Class   abbreviatedIRI="expression:Qualification"/>
</Declaration>

    <Declaration>                       <Class   abbreviatedIRI="expression:Qualified"/>
</Declaration>

    <Declaration>               <Class  abbreviatedIRI="legal-action:Legislative_Body"/>
</Declaration>

    <Declaration>                   <Class   abbreviatedIRI="legal-action:Public_Body"/>
</Declaration>

    <Declaration>       <Class abbreviatedIRI="norm:Contract"/>    </Declaration>

    <Declaration>       <Class abbreviatedIRI="norm:Legal_Document"/>    </Declaration>

    <Declaration>                      <Class   abbreviatedIRI="norm:Legal_Expression"/>
</Declaration>

    <Declaration>       <Class abbreviatedIRI="norm:Norm"/>    </Declaration>

    <Declaration>                 <Class   abbreviatedIRI="norm:Normatively_Qualified"/>
</Declaration>

    <Declaration>       <Class abbreviatedIRI="norm:Precedent"/>    </Declaration>

    <Declaration>       <Class IRI="#Adjudication"/>    </Declaration>

    <Declaration>       <Class IRI="#Agreement"/>    </Declaration>

    <Declaration>       <Class IRI="#Contractual_Agreement"/>    </Declaration>

    <Declaration>       <Class IRI="#Judgement"/>    </Declaration>

    <Declaration>       <Class IRI="#Judicial_Claim"/>    </Declaration>
```

```
    <Declaration>        <Class IRI="#Judicial_Interpretation"/>    </Declaration>

    <Declaration>        <Class IRI="#Judicial_Outcome"/>    </Declaration>

    <Declaration>        <Class IRI="#Judicially_Qualified"/>    </Declaration>

    <Declaration>        <Class IRI="#Jurisdiction"/>    </Declaration>

    <Declaration>        <Class IRI="#Law_Declaration"/>    </Declaration>

    <Declaration>        <Class IRI="#Legal_Consequence"/>        </Declaration>

    <Declaration>        <Class IRI="#Legal_Rule"/>    </Declaration>

    <Declaration>        <Class IRI="#Legal_Status"/>    </Declaration>

    <Declaration>        <Class IRI="#Material_Circumstance"/>    </Declaration>

    <Declaration>                <ObjectProperty  abbreviatedIRI="expression:attitude"/>
</Declaration>

    <Declaration>            <ObjectProperty  abbreviatedIRI="expression:evaluated_by"/>
</Declaration>

    <Declaration>               <ObjectProperty  abbreviatedIRI="expression:evaluates"/>
</Declaration>

    <Declaration>                 <ObjectProperty  abbreviatedIRI="expression:held_by"/>
</Declaration>

    <Declaration>                  <ObjectProperty  abbreviatedIRI="expression:medium"/>
</Declaration>

    <Declaration>            <ObjectProperty  abbreviatedIRI="expression:qualified_by"/>
</Declaration>

    <Declaration>               <ObjectProperty  abbreviatedIRI="expression:qualifies"/>
</Declaration>

    <Declaration>                    <ObjectProperty  abbreviatedIRI="norm:allowed_by"/>
</Declaration>

    <Declaration>                        <ObjectProperty  abbreviatedIRI="norm:allows"/>
</Declaration>

    <Declaration>                 <ObjectProperty  abbreviatedIRI="norm:disallowed_by"/>
</Declaration>

    <Declaration>                     <ObjectProperty  abbreviatedIRI="norm:disallows"/>
</Declaration>

    <EquivalentClasses>

        <Class IRI="#Judicially_Qualified"/>

        <ObjectSomeValuesFrom>

            <ObjectProperty IRI="#considered_by"/>

            <Class IRI="#Judgement"/>

        </ObjectSomeValuesFrom>

    </EquivalentClasses>
```

```
<SubClassOf>
    <Class IRI="#Adjudication"/>
    <Class IRI="#Judgement"/>
</SubClassOf>
<SubClassOf>
    <Class IRI="#Adjudication"/>
    <ObjectSomeValuesFrom>
        <ObjectProperty IRI="#applies"/>
        <Class IRI="#Judicial_Outcome"/>
    </ObjectSomeValuesFrom>
</SubClassOf>
<SubClassOf>
    <Class IRI="#Adjudication"/>
    <ObjectSomeValuesFrom>
        <ObjectProperty IRI="#considers"/>
        <Class IRI="#Judicial_Claim"/>
    </ObjectSomeValuesFrom>
</SubClassOf>
<SubClassOf>
    <Class IRI="#Agreement"/>
    <Class abbreviatedIRI="expression:Communicated_Attitude"/>
</SubClassOf>
<SubClassOf>
    <Class IRI="#Contractual_Agreement"/>
    <Class abbreviatedIRI="expression:Expression"/>
</SubClassOf>
<SubClassOf>
    <Class IRI="#Contractual_Agreement"/>
    <Class abbreviatedIRI="norm:Norm"/>
</SubClassOf>
<SubClassOf>
    <Class IRI="#Contractual_Agreement"/>
    <ObjectSomeValuesFrom>
```

```
        <ObjectProperty IRI="#applies"/>
        <Class IRI="#Legal_Status"/>
    </ObjectSomeValuesFrom>
</SubClassOf>
<SubClassOf>
    <Class IRI="#Contractual_Agreement"/>
    <ObjectSomeValuesFrom>
        <ObjectProperty IRI="#considers"/>
        <Class IRI="#Material_Circumstance"/>
    </ObjectSomeValuesFrom>
</SubClassOf>
<SubClassOf>
    <Class IRI="#Contractual_Agreement"/>
    <ObjectSomeValuesFrom>
        <ObjectProperty abbreviatedIRI="expression:attitude"/>
        <Class IRI="#Agreement"/>
    </ObjectSomeValuesFrom>
</SubClassOf>
<SubClassOf>
    <Class IRI="#Contractual_Agreement"/>
    <ObjectSomeValuesFrom>
        <ObjectProperty abbreviatedIRI="expression:medium"/>
        <Class abbreviatedIRI="norm:Contract"/>
    </ObjectSomeValuesFrom>
</SubClassOf>
<SubClassOf>
    <Class IRI="#Judgement"/>
    <Class abbreviatedIRI="expression:Expression"/>
</SubClassOf>
<SubClassOf>
    <Class IRI="#Judgement"/>
    <Class abbreviatedIRI="expression:Qualification"/>
</SubClassOf>
```

```
<SubClassOf>
    <Class IRI="#Judgement"/>
    <ObjectSomeValuesFrom>
        <ObjectProperty abbreviatedIRI="expression:attitude"/>
        <Class IRI="#Jurisdiction"/>
    </ObjectSomeValuesFrom>
</SubClassOf>
<SubClassOf>
    <Class IRI="#Judgement"/>
    <ObjectSomeValuesFrom>
        <ObjectProperty abbreviatedIRI="expression:medium"/>
        <Class abbreviatedIRI="norm:Precedent"/>
    </ObjectSomeValuesFrom>
</SubClassOf>
<SubClassOf>
    <Class IRI="#Judicial_Claim"/>
    <Class IRI="#Judicially_Qualified"/>
</SubClassOf>
<SubClassOf>
    <Class IRI="#Judicial_Interpretation"/>
    <Class IRI="#Judgement"/>
</SubClassOf>
<SubClassOf>
    <Class IRI="#Judicial_Interpretation"/>
    <ObjectSomeValuesFrom>
        <ObjectProperty IRI="#applies"/>
        <Class IRI="#Legal_Status"/>
    </ObjectSomeValuesFrom>
</SubClassOf>
<SubClassOf>
    <Class IRI="#Judicial_Interpretation"/>
    <ObjectSomeValuesFrom>
        <ObjectProperty IRI="#considers"/>
```

```
        <Class IRI="#Material_Circumstance"/>
    </ObjectSomeValuesFrom>
</SubClassOf>
<SubClassOf>
    <Class IRI="#Judicial_Outcome"/>
    <Class IRI="#Judicially_Qualified"/>
</SubClassOf>
<SubClassOf>
    <Class IRI="#Judicially_Qualified"/>
    <Class abbreviatedIRI="expression:Qualified"/>
</SubClassOf>
<SubClassOf>
    <Class IRI="#Jurisdiction"/>
    <Class abbreviatedIRI="expression:Declaration"/>
</SubClassOf>
<SubClassOf>
    <Class IRI="#Jurisdiction"/>
    <ObjectAllValuesFrom>
        <ObjectProperty abbreviatedIRI="expression:held_by"/>
        <Class abbreviatedIRI="legal-action:Public_Body"/>
    </ObjectAllValuesFrom>
</SubClassOf>
<SubClassOf>
    <Class IRI="#Law_Declaration"/>
    <Class abbreviatedIRI="expression:Declaration"/>
</SubClassOf>
<SubClassOf>
    <Class IRI="#Law_Declaration"/>
    <ObjectAllValuesFrom>
        <ObjectProperty abbreviatedIRI="expression:held_by"/>
        <Class abbreviatedIRI="legal-action:Legislative_Body"/>
    </ObjectAllValuesFrom>
</SubClassOf>
```

```
<SubClassOf>
    <Class IRI="#Legal_Consequence"/>
    <Class abbreviatedIRI="norm:Normatively_Qualified"/>
</SubClassOf>
<SubClassOf>
    <Class IRI="#Legal_Rule"/>
    <Class abbreviatedIRI="norm:Legal_Expression"/>
</SubClassOf>
<SubClassOf>
    <Class IRI="#Legal_Rule"/>
    <Class abbreviatedIRI="norm:Norm"/>
</SubClassOf>
<SubClassOf>
    <Class IRI="#Legal_Rule"/>
    <ObjectSomeValuesFrom>
        <ObjectProperty IRI="#applies"/>
        <Class IRI="#Legal_Consequence"/>
    </ObjectSomeValuesFrom>
</SubClassOf>
<SubClassOf>
    <Class IRI="#Legal_Rule"/>
    <ObjectSomeValuesFrom>
        <ObjectProperty IRI="#considers"/>
        <Class IRI="#Legal_Status"/>
    </ObjectSomeValuesFrom>
</SubClassOf>
<SubClassOf>
    <Class IRI="#Legal_Rule"/>
    <ObjectAllValuesFrom>
        <ObjectProperty abbreviatedIRI="expression:attitude"/>
        <Class IRI="#Law_Declaration"/>
    </ObjectAllValuesFrom>
</SubClassOf>
```

```
<SubClassOf>
    <Class IRI="#Legal_Rule"/>
    <ObjectAllValuesFrom>
        <ObjectProperty abbreviatedIRI="expression:medium"/>
        <Class abbreviatedIRI="norm:Legal_Document"/>
    </ObjectAllValuesFrom>
</SubClassOf>
<SubClassOf>
    <Class IRI="#Legal_Status"/>
    <Class abbreviatedIRI="norm:Normatively_Qualified"/>
</SubClassOf>
<SubClassOf>
    <Class IRI="#Material_Circumstance"/>
    <Class IRI="#Judicially_Qualified"/>
</SubClassOf>
<SubClassOf>
    <Class IRI="#Material_Circumstance"/>
    <Class abbreviatedIRI="norm:Normatively_Qualified"/>
</SubClassOf>
<SubObjectPropertyOf>
    <ObjectProperty IRI="#applied_by"/>
    <ObjectProperty abbreviatedIRI="expression:qualified_by"/>
</SubObjectPropertyOf>
<SubObjectPropertyOf>
    <ObjectProperty IRI="#applies"/>
    <ObjectProperty abbreviatedIRI="expression:qualifies"/>
</SubObjectPropertyOf>
<SubObjectPropertyOf>
    <ObjectProperty IRI="#considered_by"/>
    <ObjectProperty abbreviatedIRI="expression:qualified_by"/>
</SubObjectPropertyOf>
<SubObjectPropertyOf>
    <ObjectProperty IRI="#considers"/>
```

```
        <ObjectProperty abbreviatedIRI="expression:qualifies"/>
    </SubObjectPropertyOf>
    <SubObjectPropertyOf>
        <ObjectProperty abbreviatedIRI="expression:evaluated_by"/>
        <ObjectProperty IRI="#considered_by"/>
    </SubObjectPropertyOf>
    <SubObjectPropertyOf>
        <ObjectProperty abbreviatedIRI="expression:evaluates"/>
        <ObjectProperty IRI="#considers"/>
    </SubObjectPropertyOf>
    <SubObjectPropertyOf>
        <ObjectProperty abbreviatedIRI="norm:allowed_by"/>
        <ObjectProperty IRI="#considered_by"/>
    </SubObjectPropertyOf>
    <SubObjectPropertyOf>
        <ObjectProperty abbreviatedIRI="norm:allows"/>
        <ObjectProperty IRI="#considers"/>
    </SubObjectPropertyOf>
    <SubObjectPropertyOf>
        <ObjectProperty abbreviatedIRI="norm:disallowed_by"/>
        <ObjectProperty IRI="#considered_by"/>
    </SubObjectPropertyOf>
    <SubObjectPropertyOf>
        <ObjectProperty abbreviatedIRI="norm:disallows"/>
        <ObjectProperty IRI="#considers"/>
    </SubObjectPropertyOf>
    <InverseObjectProperties>
        <ObjectProperty IRI="#applied_by"/>
        <ObjectProperty IRI="#applies"/>
    </InverseObjectProperties>
    <InverseObjectProperties>
        <ObjectProperty IRI="#considers"/>
        <ObjectProperty IRI="#considered_by"/>
```

```
    </InverseObjectProperties>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>

        <IRI>#Adjudication</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">The final assessment of a judge on a claim.</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:label"/>

        <IRI>#Adjudication</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Adjudication</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>

        <IRI>#Agreement</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">The empowering attitude of contractual agreements.</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:label"/>

        <IRI>#Agreement</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Agreement</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>

        <IRI>#Contractual_Agreement</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">The content of a contract. Takes into consideration a specific Material circumstance and applies a legal status to it.</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:label"/>

        <IRI>#Contractual_Agreement</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Contractual Agreement</Literal>

    </AnnotationAssertion>
```

```
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#Judgement</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">The syntactic representation of a judicial decision&#39;s contents</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:label"/>
        <IRI>#Judgement</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Judgement</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#Judicial_Claim</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Is the claim brought in front of the judge by one of the parties.</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:label"/>
        <IRI>#Judicial_Claim</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Judicial Claim</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#Judicial_Interpretation</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Judicial interpretations are performative expressions contained in sentences (i.e. subsumptions). They take into consideration a Factual circumstance and apply a legal status to it.</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:label"/>
        <IRI>#Judicial_Interpretation</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Judicial Interpretation</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
```

```
        <AnnotationProperty abbreviatedIRI="rdfs:label"/>

        <IRI>#Judicial_Outcome</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Judicial Outcome</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="owl:incompatibleWith"/>

        <IRI>#Judicial_Outcome</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Is the outcome applied by the judge to
every judicial claim. i.e. &quot;accepted&quot;, &quot;rejected&quot;</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>

        <IRI>#Judicially_Qualified</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Anything that is qualified by a
judicial expression.</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:label"/>

        <IRI>#Judicially_Qualified</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Judicially Qualified</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>

        <IRI>#Jurisdiction</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">The empowering attitude of
judgements</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:label"/>

        <IRI>#Jurisdiction</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Jurisdiction</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
```

```
        <IRI>#Law_Declaration</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">The empowering attitude of legal rules</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:label"/>

        <IRI>#Law_Declaration</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Law Declaration</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>

        <IRI>#Legal_Consequence</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">A legal consequence is the consequent of a legal rule. In the form of regulative rules &quot;if A then B&quot;, B is the Legal Consequence. i.e. a sanction.</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:label"/>

        <IRI>#Legal_Consequence</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Legal Consequence</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>

        <IRI>#Legal_Rule</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">The content of a legal document. considers anything having a given (combination of) legal status(es) and applies a (combination of) legal consequence(s) to it.</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:label"/>

        <IRI>#Legal_Rule</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Legal Rule</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
```

```
        <IRI>#Legal_Status</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">A legal status is an abstract concept which is created by a law (or more rarely by doctrine or customs). It can be taken by a judgement or a contract and applied to some Factual circumstance, more or less arbitrarily. I.e. &quot;knowable&quot; is a legal status.</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:label"/>

        <IRI>#Legal_Status</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Legal Status</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>

        <IRI>#Material_Circumstance</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">a Factual Circumstance is any fact or act which occurred in the Factual world, and which is taken into consideration by a law, a contract, or a judgement.</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:label"/>

        <IRI>#Material_Circumstance</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Material Circumstance</Literal>

    </AnnotationAssertion>

</Ontology>




<!-- Generated by the OWL API (version 3.2.3.1824) http://owlapi.sourceforge.net -->
```

a.1.2. Domain ontology

```
<?xml version="1.0"?>



<!DOCTYPE Ontology [

    <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >

    <!ENTITY xml "http://www.w3.org/XML/1998/namespace" >
```

```
    &lt;!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" &gt;

    &lt;!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" &gt;

]>



<Ontology xmlns="http://www.w3.org/2002/07/owl#"

     xml:base="http://codexml.cirsfid.unibo.it/ontologies/Judging_Contracts_Domain.owl"

     xmlns:xsd="http://www.w3.org/2001/XMLSchema#"

     xmlns:xml="http://www.w3.org/XML/1998/namespace"

     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"

     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"


ontologyIRI="http://codexml.cirsfid.unibo.it/ontologies/Judging_Contracts_Domain.owl">

    &lt;Prefix name="mereology" IRI="http://www.estrellaproject.org/lkif-core/mereology.owl#"/&gt;

    &lt;Prefix name="rdf" IRI="http://www.w3.org/1999/02/22-rdf-syntax-ns#"/&gt;

    &lt;Prefix name="norm" IRI="http://www.estrellaproject.org/lkif-core/norm.owl#"/&gt;

    &lt;Prefix name="" IRI="http://www.w3.org/2002/07/owl#"/&gt;

    &lt;Prefix name="xsd" IRI="http://www.w3.org/2001/XMLSchema#"/&gt;

    &lt;Prefix name="Judging_Contracts_Domain" IRI="http://codexml.cirsfid.unibo.it/ontologies/Judging_Contracts_Domain.owl#"/&gt;

    &lt;Prefix name="expression" IRI="http://www.estrellaproject.org/lkif-core/expression.owl#"/&gt;

    &lt;Prefix name="Judging_Contracts_Core" IRI="http://codexml.cirsfid.unibo.it/ontologies/Judging_Contracts_Core.owl#"/&gt;

    &lt;Prefix name="rdfs" IRI="http://www.w3.org/2000/01/rdf-schema#"/&gt;

    &lt;Prefix name="owl" IRI="http://www.w3.org/2002/07/owl#"/&gt;

    &lt;Declaration&gt; &lt;Class abbreviatedIRI="Judging_Contracts_Core:Adjudication"/&gt; &lt;/Declaration&gt;

    &lt;Declaration&gt; &lt;Class abbreviatedIRI="Judging_Contracts_Core:Agreement"/&gt; &lt;/Declaration&gt;

    &lt;Declaration&gt; &lt;Class abbreviatedIRI="Judging_Contracts_Core:Contractual_Agreement"/&gt; &lt;/Declaration&gt;

    &lt;Declaration&gt; &lt;Class abbreviatedIRI="Judging_Contracts_Core:Judgement"/&gt; &lt;/Declaration&gt;

    &lt;Declaration&gt; &lt;Class abbreviatedIRI="Judging_Contracts_Core:Judicial_Claim"/&gt; &lt;/Declaration&gt;
```

```
    <Declaration>                                                          <Class
abbreviatedIRI="Judging_Contracts_Core:Judicial_Interpretation"/>    </Declaration>

    <Declaration>      <Class  abbreviatedIRI="Judging_Contracts_Core:Judicial_Outcome"/>
</Declaration>

    <Declaration>                                                          <Class
abbreviatedIRI="Judging_Contracts_Core:Judicially_Qualified"/>  </Declaration>

    <Declaration>        <Class   abbreviatedIRI="Judging_Contracts_Core:Jurisdiction"/>
</Declaration>

    <Declaration>      <Class  abbreviatedIRI="Judging_Contracts_Core:Law_Declaration"/>
</Declaration>

    <Declaration>     <Class  abbreviatedIRI="Judging_Contracts_Core:Legal_Consequence"/>
</Declaration>

    <Declaration>         <Class    abbreviatedIRI="Judging_Contracts_Core:Legal_Rule"/>
</Declaration>

    <Declaration>        <Class   abbreviatedIRI="Judging_Contracts_Core:Legal_Status"/>
</Declaration>

    <Declaration>                                                          <Class
abbreviatedIRI="Judging_Contracts_Core:Material_Circumstance"/>  </Declaration>

    <Declaration>   <Class IRI="#Art1341-1342_Exception"/>    </Declaration>

    <Declaration>   <Class IRI="#Contract_Clause"/>   </Declaration>

    <Declaration>   <Class IRI="#Efficacious_ViaException"/>    </Declaration>

    <Declaration>   <Class IRI="#Exception"/>    </Declaration>

    <Declaration>   <Class IRI="#Full_Contract"/>    </Declaration>

    <Declaration>   <Class IRI="#Inefficacious"/>    </Declaration>

    <Declaration>   <Class IRI="#Inefficacy"/>    </Declaration>

    <Declaration>   <Class IRI="#NonOppressive_Status"/>    </Declaration>

    <Declaration>   <Class IRI="#Oppressive_Status"/>    </Declaration>

    <Declaration>   <Class IRI="#Relevant_ExArt1341co1"/>    </Declaration>

    <Declaration>   <Class IRI="#Relevant_ExArt1341co2"/>    </Declaration>

    <Declaration>   <Class IRI="#Relevant_ExArt1342co1"/>    </Declaration>

    <Declaration>   <Class IRI="#Relevant_ExArt1342co2"/>    </Declaration>

    <Declaration>   <Class IRI="#Reproducing_LawDisposition"/>    </Declaration>

    <Declaration>   <Class IRI="#Valid_ExArt1341co1"/>    </Declaration>

    <Declaration>   <Class IRI="#Valid_ExArt1341co2"/>    </Declaration>

    <Declaration>          <Class    abbreviatedIRI="expression:Communicated_Attitude"/>
</Declaration>

    <Declaration>   <Class abbreviatedIRI="expression:Declaration"/>    </Declaration>

    <Declaration>   <Class abbreviatedIRI="expression:Expression"/>    </Declaration>
```

```
<Declaration>   <Class abbreviatedIRI="expression:Qualification"/>    </Declaration>

<Declaration>   <Class abbreviatedIRI="expression:Qualified"/>    </Declaration>

<Declaration>   <Class abbreviatedIRI="norm:Legal_Expression"/>    </Declaration>

<Declaration>   <Class abbreviatedIRI="norm:Norm"/>    </Declaration>

<Declaration>   <Class abbreviatedIRI="norm:Normatively_Qualified"/>    </Declaration>

<Declaration>   <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applied_by"/>    </Declaration>

<Declaration>   <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/>    </Declaration>

<Declaration>   <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:considered_by"/>    </Declaration>

<Declaration>   <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:considers"/>    </Declaration>

<Declaration>   <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:judged_as"/>    </Declaration>

<Declaration>   <ObjectProperty IRI="#assumption"/>    </Declaration>

<Declaration>   <ObjectProperty IRI="#exception"/>    </Declaration>

<Declaration>   <ObjectProperty abbreviatedIRI="expression:attitude"/>    </Declaration>

<Declaration>   <ObjectProperty abbreviatedIRI="expression:evaluated_by"/>    </Declaration>

<Declaration>   <ObjectProperty abbreviatedIRI="expression:evaluates"/>    </Declaration>

<Declaration>   <ObjectProperty abbreviatedIRI="expression:qualified_by"/>    </Declaration>

<Declaration>   <ObjectProperty abbreviatedIRI="expression:qualifies"/>    </Declaration>

<Declaration>   <ObjectProperty abbreviatedIRI="mereology:contained_in"/>    </Declaration>

<Declaration>   <ObjectProperty abbreviatedIRI="norm:allowed_by"/>    </Declaration>

<Declaration>   <ObjectProperty abbreviatedIRI="norm:allows"/>    </Declaration>

<Declaration>   <ObjectProperty abbreviatedIRI="norm:disallowed_by"/>    </Declaration>

<Declaration>   <ObjectProperty abbreviatedIRI="norm:disallows"/>    </Declaration>

<Declaration>   <NamedIndividual IRI="#AOUMessina_ByteSoftwareHouse_Clause2.020"/>    </Declaration>

<Declaration>   <NamedIndividual IRI="#AcquedottoPugliese_CC_Clause8"/>    </Declaration>

<Declaration>   <NamedIndividual IRI="#AddedToPrecompiled"/>    </Declaration>
```

```
    <Declaration>        <NamedIndividual   IRI="#AgenziaImmobiliareD_NG&amp;PF_ClauseX"/>
</Declaration>

    <Declaration>   <NamedIndividual IRI="#Alfa_Beta1_Clause8"/>    </Declaration>

    <Declaration>   <NamedIndividual IRI="#Arbitration"/>    </Declaration>

    <Declaration>   <NamedIndividual IRI="#Art1219co3cc"/>    </Declaration>

    <Declaration>   <NamedIndividual IRI="#Art1341co1cc"/>    </Declaration>

    <Declaration>   <NamedIndividual IRI="#Art1341co2cc"/>    </Declaration>

    <Declaration>   <NamedIndividual IRI="#Art1342co1cc"/>    </Declaration>

    <Declaration>   <NamedIndividual IRI="#Art1342co2cc"/>    </Declaration>

    <Declaration>   <NamedIndividual IRI="#Art1743cc"/>    </Declaration>

    <Declaration>                                                       <NamedIndividual
IRI="#AssicurazioniGenerali_Csas_SectionA_Clause2"/>    </Declaration>

    <Declaration>       <NamedIndividual   IRI="#AutomobilClubEuropeo_GeCapital_ClauseX"/>
</Declaration>

    <Declaration>          <NamedIndividual    IRI="#BC_BancaCreditoCooperativo_Clause6"/>
</Declaration>

    <Declaration>   <NamedIndividual IRI="#BL_Assitalia_Clause2.8"/>    </Declaration>

    <Declaration>            <NamedIndividual    IRI="#CG_CarigeAssicurazioni_Clause31"/>
</Declaration>

    <Declaration>   <NamedIndividual IRI="#CG_IntesaVita_Clause6"/>    </Declaration>

    <Declaration>   <NamedIndividual IRI="#CdABariII_Int1"/>    </Declaration>

    <Declaration>   <NamedIndividual IRI="#CdAFirenzeI_Int1"/>    </Declaration>

    <Declaration>   <NamedIndividual IRI="#CdAMilanoII_Int1"/>    </Declaration>

    <Declaration>   <NamedIndividual IRI="#CdARomaII_Int1"/>    </Declaration>

    <Declaration>   <NamedIndividual IRI="#CompetenceDerogation"/>    </Declaration>

    <Declaration>          <NamedIndividual    IRI="#ComunePortoAzzurro_Daneco_Clause21"/>
</Declaration>

    <Declaration>         <NamedIndividual    IRI="#CondominioViaDelGambero_BM_Clause16"/>
</Declaration>

    <Declaration>       <NamedIndividual   IRI="#CondominioViaRasori_Santacroce_ClauseX"/>
</Declaration>

    <Declaration>           <NamedIndividual    IRI="#ContractWithdrawalForBothParties"/>
</Declaration>

    <Declaration>              <NamedIndividual     IRI="#ContractWithdrawalUnilateral"/>
</Declaration>

    <Declaration>    <NamedIndividual  IRI="#EuropaPalaceHotel_SeatPagineGialle_Clause4"/>
</Declaration>

    <Declaration>   <NamedIndividual IRI="#Exception_Limitation"/>    </Declaration>
```

```
<Declaration>  <NamedIndividual IRI="#ExpertOpinion"/>   </Declaration>

<Declaration>  <NamedIndividual IRI="#General"/>   </Declaration>

<Declaration>           <NamedIndividual    IRI="#IBIImmobiliare_AACF_ClauseC"/>
</Declaration>

<Declaration>  <NamedIndividual IRI="#ISPB_SAI_clause_12"/>   </Declaration>

<Declaration>  <NamedIndividual IRI="#Inefficacy_ExArt1341co1"/>  </Declaration>

<Declaration>  <NamedIndividual IRI="#Inefficacy_ExArt1341co2"/>   </Declaration>

<Declaration>  <NamedIndividual IRI="#Inefficacy_ExArt1342co2"/>   </Declaration>

<Declaration>  <NamedIndividual IRI="#International"/>   </Declaration>

<Declaration>  <NamedIndividual IRI="#Knowable"/>   </Declaration>

<Declaration>  <NamedIndividual IRI="#LLoyd_FF_Clause*"/>   </Declaration>

<Declaration>  <NamedIndividual IRI="#LLoyd_FF_Contract"/>   </Declaration>

<Declaration>  <NamedIndividual IRI="#Law180_1993Art2co2"/>   </Declaration>

<Declaration>  <NamedIndividual IRI="#LiabilityLimitation"/>   </Declaration>

<Declaration>           <NamedIndividual    IRI="#LimitationTowards3rdParties"/>
</Declaration>

<Declaration>  <NamedIndividual IRI="#ME_LaSorgente_Clause8"/>   </Declaration>

<Declaration>           <NamedIndividual    IRI="#MeAComputer_Cartasi_ClauseX"/>
</Declaration>

<Declaration>           <NamedIndividual    IRI="#NewEdge_FotoVillage_clause12"/>
</Declaration>

<Declaration>  <NamedIndividual IRI="#No_Arbitration"/>   </Declaration>

<Declaration>  <NamedIndividual IRI="#No_CompentenceDerogation"/>   </Declaration>

<Declaration>  <NamedIndividual IRI="#No_ContractWithdrawal"/>   </Declaration>

<Declaration>           <NamedIndividual    IRI="#No_ResponsibilityLimitation"/>
</Declaration>

<Declaration>  <NamedIndividual IRI="#NotSpecificallySigned"/>   </Declaration>

<Declaration>  <NamedIndividual IRI="#Not_Knowable"/>   </Declaration>

<Declaration>  <NamedIndividual IRI="#Not_Unilateral"/>   </Declaration>

<Declaration>    <NamedIndividual  IRI="#NovusWohnbedarfVertriebs_Bspa_Clause3.1"/>
</Declaration>

<Declaration>  <NamedIndividual IRI="#Precompiled"/>   </Declaration>

<Declaration>          <NamedIndividual    IRI="#PrevailsOverPrecompiledClauses"/>
</Declaration>

<Declaration>            <NamedIndividual     IRI="#ReproducingLawDisposition"/>
<Deeclaration>

<Declaration>  <NamedIndividual IRI="#ResponsibilityExtension"/>   </Declaration>
```

```
    <Declaration>   <NamedIndividual IRI="#ResponsibilityLimitation"/>    </Declaration>

    <Declaration>      <NamedIndividual  IRI="#SC_LaPiemonteseAssicurazioni_Clause15f"/>
</Declaration>

    <Declaration>   <NamedIndividual IRI="#STF_KSBClause10.2"/>    </Declaration>

    <Declaration>   <NamedIndividual IRI="#Specific"/>    </Declaration>

    <Declaration>   <NamedIndividual IRI="#SpecificallySigned"/>    </Declaration>

    <Declaration>   <NamedIndividual IRI="#StatutoFIGC_Clause24"/>    </Declaration>

    <Declaration>                                                        <NamedIndividual
IRI="#ToroAssicurazioni_ImpresaAutoimpianti_ClauseB"/>    </Declaration>

    <Declaration>   <NamedIndividual IRI="#TribBariIII_Int1"/>    </Declaration>

    <Declaration>   <NamedIndividual IRI="#TribComoI_Int1"/>    </Declaration>

    <Declaration>   <NamedIndividual IRI="#TribFirenzeIII_Int1"/>    </Declaration>

    <Declaration>   <NamedIndividual IRI="#TribGenovaI_Int1"/>    </Declaration>

    <Declaration>   <NamedIndividual IRI="#TribIsernia_int1"/>    </Declaration>

    <Declaration>   <NamedIndividual IRI="#TribIsernia_int1.2"/>    </Declaration>

    <Declaration>   <NamedIndividual IRI="#TribModenaII_Int1"/>    </Declaration>

    <Declaration>   <NamedIndividual IRI="#TribMondovi_Int1"/>    </Declaration>

    <Declaration>   <NamedIndividual IRI="#TribMonzaI_Int1"/>    </Declaration>

    <Declaration>   <NamedIndividual IRI="#TribMonzaI_Int2.1"/>    </Declaration>

    <Declaration>   <NamedIndividual IRI="#TribNapoliI_Int1"/>    </Declaration>

    <Declaration>   <NamedIndividual IRI="#TribNolaII_Int1"/>    </Declaration>

    <Declaration>   <NamedIndividual IRI="#TribPiacenzaI_Int1"/>    </Declaration>

    <Declaration>   <NamedIndividual IRI="#TribPiacenzaI_Int2.1"/>    </Declaration>

    <Declaration>   <NamedIndividual IRI="#TribRomaIX_Int1"/>    </Declaration>

    <Declaration>   <NamedIndividual IRI="#TribSalernoI_Int1"/>    </Declaration>

    <Declaration>   <NamedIndividual IRI="#TribSavonaI_Int1"/>    </Declaration>

    <Declaration>   <NamedIndividual IRI="#TribTorinoIII_Int1"/>    </Declaration>

    <Declaration>   <NamedIndividual IRI="#TribTorinoIII_Int2.1"/>    </Declaration>

    <Declaration>   <NamedIndividual IRI="#TribTorinoI_Int1"/>    </Declaration>

    <Declaration>   <NamedIndividual IRI="#TribTorinoVIII_Int1"/>    </Declaration>

    <Declaration>   <NamedIndividual IRI="#TribTraniI_Int1"/>    </Declaration>

    <Declaration>   <NamedIndividual IRI="#TribVeneziaI_Int1"/>    </Declaration>

    <Declaration>   <NamedIndividual IRI="#Unilateral"/>    </Declaration>

    <Declaration>   <NamedIndividual IRI="#VF_BPTR_clauseX"/>    </Declaration>
```

```xml
<EquivalentClasses>
    <Class IRI="#Contract_Clause"/>
    <ObjectIntersectionOf>
        <Class abbreviatedIRI="Judging_Contracts_Core:Contractual_Agreement"/>
        <ObjectSomeValuesFrom>
            <ObjectProperty abbreviatedIRI="mereology:contained_in"/>
            <Class IRI="#Full_Contract"/>
        </ObjectSomeValuesFrom>
    </ObjectIntersectionOf>
</EquivalentClasses>
<EquivalentClasses>
    <Class IRI="#Efficacious_ViaException"/>
    <ObjectIntersectionOf>
        <Class abbreviatedIRI="Judging_Contracts_Core:Contractual_Agreement"/>
        <Class IRI="#Inefficacious"/>
        <ObjectUnionOf>
            <ObjectSomeValuesFrom>
                <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/>
                <Class IRI="#Art1341-1342_Exception"/>
            </ObjectSomeValuesFrom>
            <ObjectSomeValuesFrom>
                <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:judged_as"/>
                <Class IRI="#Art1341-1342_Exception"/>
            </ObjectSomeValuesFrom>
        </ObjectUnionOf>
    </ObjectIntersectionOf>
</EquivalentClasses>
<EquivalentClasses>
    <Class IRI="#Inefficacious"/>
    <ObjectIntersectionOf>
        <Class abbreviatedIRI="Judging_Contracts_Core:Contractual_Agreement"/>
        <ObjectSomeValuesFrom>
            <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:judged_as"/>
```

```
            <Class IRI="#Inefficacy"/>
        </ObjectSomeValuesFrom>
    </ObjectIntersectionOf>
</EquivalentClasses>
<EquivalentClasses>
    <Class IRI="#NonOppressive_Status"/>
    <ObjectIntersectionOf>
        <Class abbreviatedIRI="Judging_Contracts_Core:Legal_Status"/>
        <ObjectSomeValuesFrom>
            <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:judged_as"/>
            <Class IRI="#NonOppressive_Status"/>
        </ObjectSomeValuesFrom>
    </ObjectIntersectionOf>
</EquivalentClasses>
<EquivalentClasses>
    <Class IRI="#Relevant_ExArt1341co1"/>
    <ObjectIntersectionOf>
        <Class abbreviatedIRI="Judging_Contracts_Core:Contractual_Agreement"/>
        <ObjectUnionOf>
            <ObjectHasValue>
                <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/>
                <NamedIndividual IRI="#General"/>
            </ObjectHasValue>
            <ObjectHasValue>
                <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:judged_as"/>
                <NamedIndividual IRI="#General"/>
            </ObjectHasValue>
        </ObjectUnionOf>
        <ObjectUnionOf>
            <ObjectHasValue>
                <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/>
                <NamedIndividual IRI="#Not_Knowable"/>
            </ObjectHasValue>
```

```
            <ObjectHasValue>
                <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:judged_as"/>
                <NamedIndividual IRI="#Not_Knowable"/>
            </ObjectHasValue>
        </ObjectUnionOf>
        <ObjectUnionOf>
            <ObjectHasValue>
                <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/>
                <NamedIndividual IRI="#Unilateral"/>
            </ObjectHasValue>
            <ObjectHasValue>
                <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:judged_as"/>
                <NamedIndividual IRI="#Unilateral"/>
            </ObjectHasValue>
        </ObjectUnionOf>
    </ObjectIntersectionOf>
</EquivalentClasses>
<EquivalentClasses>
    <Class IRI="#Relevant_ExArt1341co1"/>
    <ObjectHasValue>
        <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:considered_by"/>
        <NamedIndividual IRI="#Art1341co1cc"/>
    </ObjectHasValue>
</EquivalentClasses>
<EquivalentClasses>
    <Class IRI="#Relevant_ExArt1341co2"/>
    <ObjectIntersectionOf>
        <Class abbreviatedIRI="Judging_Contracts_Core:Contractual_Agreement"/>
        <ObjectUnionOf>
            <ObjectSomeValuesFrom>
                <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/>
                <Class IRI="#Oppressive_Status"/>
            </ObjectSomeValuesFrom>
```

```xml
    <ObjectSomeValuesFrom>
        <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:judged_as"/>
        <Class IRI="#Oppressive_Status"/>
    </ObjectSomeValuesFrom>
</ObjectUnionOf>
<ObjectUnionOf>
    <ObjectHasValue>
        <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/>
        <NamedIndividual IRI="#General"/>
    </ObjectHasValue>
    <ObjectHasValue>
        <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:judged_as"/>
        <NamedIndividual IRI="#General"/>
    </ObjectHasValue>
</ObjectUnionOf>
<ObjectUnionOf>
    <ObjectHasValue>
        <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/>
        <NamedIndividual IRI="#NotSpecificallySigned"/>
    </ObjectHasValue>
    <ObjectHasValue>
        <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:judged_as"/>
        <NamedIndividual IRI="#NotSpecificallySigned"/>
    </ObjectHasValue>
</ObjectUnionOf>
<ObjectUnionOf>
    <ObjectHasValue>
        <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/>
        <NamedIndividual IRI="#Unilateral"/>
    </ObjectHasValue>
    <ObjectHasValue>
        <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:judged_as"/>
        <NamedIndividual IRI="#Unilateral"/>
```

```
                </ObjectHasValue>
            </ObjectUnionOf>
        </ObjectIntersectionOf>
    </EquivalentClasses>
    <EquivalentClasses>
        <Class IRI="#Relevant_ExArt1341co2"/>
        <ObjectHasValue>
            <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:considered_by"/>
            <NamedIndividual IRI="#Art1341co2cc"/>
        </ObjectHasValue>
    </EquivalentClasses>
    <EquivalentClasses>
        <Class IRI="#Relevant_ExArt1342co1"/>
        <ObjectIntersectionOf>
            <Class abbreviatedIRI="Judging_Contracts_Core:Contractual_Agreement"/>
            <ObjectUnionOf>
                <ObjectHasValue>
                    <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/>
                    <NamedIndividual IRI="#AddedToPrecompiled"/>
                </ObjectHasValue>
                <ObjectHasValue>
                    <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:judged_as"/>
                    <NamedIndividual IRI="#AddedToPrecompiled"/>
                </ObjectHasValue>
            </ObjectUnionOf>
            <ObjectSomeValuesFrom>
                <ObjectProperty abbreviatedIRI="mereology:contained_in"/>
                <ObjectUnionOf>
                    <ObjectHasValue>
                        <ObjectProperty
abbreviatedIRI="Judging_Contracts_Core:applies"/>
                        <NamedIndividual IRI="#Precompiled"/>
                    </ObjectHasValue>
```

```
                    <ObjectHasValue>
                        <ObjectProperty
abbreviatedIRI="Judging_Contracts_Core:judged_as"/>
                        <NamedIndividual IRI="#Precompiled"/>
                    </ObjectHasValue>
                </ObjectUnionOf>
            </ObjectSomeValuesFrom>
        </ObjectIntersectionOf>
    </EquivalentClasses>
    <EquivalentClasses>
        <Class IRI="#Relevant_ExArt1342co1"/>
        <ObjectHasValue>
            <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:considered_by"/>
            <NamedIndividual IRI="#Art1342co1cc"/>
        </ObjectHasValue>
    </EquivalentClasses>
    <EquivalentClasses>
        <Class IRI="#Relevant_ExArt1342co2"/>
        <ObjectIntersectionOf>
            <Class abbreviatedIRI="Judging_Contracts_Core:Contractual_Agreement"/>
            <Class IRI="#Relevant_ExArt1342co1"/>
            <ObjectUnionOf>
                <ObjectHasValue>
                    <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/>
                    <NamedIndividual IRI="#NotSpecificallySigned"/>
                </ObjectHasValue>
                <ObjectHasValue>
                    <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:judged_as"/>
                    <NamedIndividual IRI="#NotSpecificallySigned"/>
                </ObjectHasValue>
            </ObjectUnionOf>
        </ObjectIntersectionOf>
    </EquivalentClasses>
```

```xml
<EquivalentClasses>
    <Class IRI="#Relevant_ExArt1342co2"/>
    <ObjectHasValue>
        <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:considered_by"/>
        <NamedIndividual IRI="#Art1342co2cc"/>
    </ObjectHasValue>
</EquivalentClasses>
<EquivalentClasses>
    <Class IRI="#Reproducing_LawDisposition"/>
    <ObjectIntersectionOf>
        <Class abbreviatedIRI="Judging_Contracts_Core:Contractual_Agreement"/>
        <ObjectUnionOf>
            <ObjectSomeValuesFrom>
                <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/>
                <Class abbreviatedIRI="Judging_Contracts_Core:Legal_Rule"/>
            </ObjectSomeValuesFrom>
            <ObjectSomeValuesFrom>
                <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:judged_as"/>
                <Class abbreviatedIRI="Judging_Contracts_Core:Legal_Rule"/>
            </ObjectSomeValuesFrom>
        </ObjectUnionOf>
        <ObjectUnionOf>
            <ObjectHasValue>
                <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/>
                <NamedIndividual IRI="#ReproducingLawDisposition"/>
            </ObjectHasValue>
            <ObjectHasValue>
                <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:judged_as"/>
                <NamedIndividual IRI="#ReproducingLawDisposition"/>
            </ObjectHasValue>
        </ObjectUnionOf>
    </ObjectIntersectionOf>
</EquivalentClasses>
```

```
    <EquivalentClasses>
        <Class IRI="#Valid_ExArt1341co1"/>
        <ObjectIntersectionOf>
            <Class abbreviatedIRI="Judging_Contracts_Core:Contractual_Agreement"/>
            <ObjectUnionOf>
                <ObjectUnionOf>
                    <ObjectHasValue>
                        <ObjectProperty
abbreviatedIRI="Judging_Contracts_Core:applies"/>
                        <NamedIndividual IRI="#Knowable"/>
                    </ObjectHasValue>
                    <ObjectHasValue>
                        <ObjectProperty
abbreviatedIRI="Judging_Contracts_Core:judged_as"/>
                        <NamedIndividual IRI="#Knowable"/>
                    </ObjectHasValue>
                </ObjectUnionOf>
                <ObjectUnionOf>
                    <ObjectHasValue>
                        <ObjectProperty
abbreviatedIRI="Judging_Contracts_Core:applies"/>
                        <NamedIndividual IRI="#Not_Unilateral"/>
                    </ObjectHasValue>
                    <ObjectHasValue>
                        <ObjectProperty
abbreviatedIRI="Judging_Contracts_Core:judged_as"/>
                        <NamedIndividual IRI="#Not_Unilateral"/>
                    </ObjectHasValue>
                </ObjectUnionOf>
                <ObjectUnionOf>
                    <ObjectHasValue>
                        <ObjectProperty
abbreviatedIRI="Judging_Contracts_Core:applies"/>
                        <NamedIndividual IRI="#Specific"/>
                    </ObjectHasValue>
```

```
                    <ObjectHasValue>
                        <ObjectProperty
abbreviatedIRI="Judging_Contracts_Core:judged_as"/>
                        <NamedIndividual IRI="#Specific"/>
                    </ObjectHasValue>
                </ObjectUnionOf>
                <ObjectUnionOf>
                    <ObjectHasValue>
                        <ObjectProperty
abbreviatedIRI="Judging_Contracts_Core:applies"/>
                        <NamedIndividual IRI="#SpecificallySigned"/>
                    </ObjectHasValue>
                    <ObjectHasValue>
                        <ObjectProperty
abbreviatedIRI="Judging_Contracts_Core:judged_as"/>
                        <NamedIndividual IRI="#SpecificallySigned"/>
                    </ObjectHasValue>
                </ObjectUnionOf>
            </ObjectUnionOf>
        </ObjectIntersectionOf>
    </EquivalentClasses>
    <EquivalentClasses>
        <Class IRI="#Valid_ExArt1341co1"/>
        <ObjectIntersectionOf>
            <Class abbreviatedIRI="Judging_Contracts_Core:Contractual_Agreement"/>
            <ObjectComplementOf>
                <Class IRI="#Relevant_ExArt1341co1"/>
            </ObjectComplementOf>
        </ObjectIntersectionOf>
    </EquivalentClasses>
    <EquivalentClasses>
        <Class IRI="#Valid_ExArt1341co2"/>
        <ObjectIntersectionOf>
            <Class abbreviatedIRI="Judging_Contracts_Core:Contractual_Agreement"/>
```

```
            <ObjectUnionOf>
                <ObjectUnionOf>
                    <ObjectSomeValuesFrom>
                        <ObjectProperty
abbreviatedIRI="Judging_Contracts_Core:applies"/>
                        <Class IRI="#NonOppressive_Status"/>
                    </ObjectSomeValuesFrom>
                    <ObjectSomeValuesFrom>
                        <ObjectProperty
abbreviatedIRI="Judging_Contracts_Core:judged_as"/>
                        <Class IRI="#NonOppressive_Status"/>
                    </ObjectSomeValuesFrom>
                </ObjectUnionOf>
                <ObjectUnionOf>
                    <ObjectHasValue>
                        <ObjectProperty
abbreviatedIRI="Judging_Contracts_Core:applies"/>
                        <NamedIndividual IRI="#Not_Unilateral"/>
                    </ObjectHasValue>
                    <ObjectHasValue>
                        <ObjectProperty
abbreviatedIRI="Judging_Contracts_Core:judged_as"/>
                        <NamedIndividual IRI="#Not_Unilateral"/>
                    </ObjectHasValue>
                </ObjectUnionOf>
                <ObjectUnionOf>
                    <ObjectHasValue>
                        <ObjectProperty
abbreviatedIRI="Judging_Contracts_Core:applies"/>
                        <NamedIndividual IRI="#Specific"/>
                    </ObjectHasValue>
                    <ObjectHasValue>
                        <ObjectProperty
abbreviatedIRI="Judging_Contracts_Core:judged_as"/>
                        <NamedIndividual IRI="#Specific"/>
                    </ObjectHasValue>
```

```
                </ObjectUnionOf>
                <ObjectUnionOf>
                    <ObjectHasValue>
                        <ObjectProperty
abbreviatedIRI="Judging_Contracts_Core:applies"/>
                        <NamedIndividual IRI="#SpecificallySigned"/>
                    </ObjectHasValue>
                    <ObjectHasValue>
                        <ObjectProperty
abbreviatedIRI="Judging_Contracts_Core:judged_as"/>
                        <NamedIndividual IRI="#SpecificallySigned"/>
                    </ObjectHasValue>
                </ObjectUnionOf>
            </ObjectUnionOf>
        </ObjectIntersectionOf>
    </EquivalentClasses>
    <EquivalentClasses>
        <Class IRI="#Valid_ExArt1341co2"/>
        <ObjectIntersectionOf>
            <Class abbreviatedIRI="Judging_Contracts_Core:Contractual_Agreement"/>
            <ObjectComplementOf>
                <Class IRI="#Relevant_ExArt1341co2"/>
            </ObjectComplementOf>
        </ObjectIntersectionOf>
    </EquivalentClasses>
    <SubClassOf>
        <Class abbreviatedIRI="Judging_Contracts_Core:Adjudication"/>
        <Class abbreviatedIRI="Judging_Contracts_Core:Judgement"/>
    </SubClassOf>
    <SubClassOf>
        <Class abbreviatedIRI="Judging_Contracts_Core:Contractual_Agreement"/>
        <ObjectSomeValuesFrom>
            <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:considers"/>
            <Class abbreviatedIRI="Judging_Contracts_Core:Material_Circumstance"/>
```

```
        </ObjectSomeValuesFrom>
    </SubClassOf>
    <SubClassOf>
        <Class abbreviatedIRI="Judging_Contracts_Core:Judicial_Interpretation"/>
        <Class abbreviatedIRI="Judging_Contracts_Core:Judgement"/>
    </SubClassOf>
    <SubClassOf>
        <Class abbreviatedIRI="Judging_Contracts_Core:Judicial_Interpretation"/>
        <ObjectSomeValuesFrom>
            <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:considers"/>
            <Class abbreviatedIRI="Judging_Contracts_Core:Contractual_Agreement"/>
        </ObjectSomeValuesFrom>
    </SubClassOf>
    <SubClassOf>
        <Class abbreviatedIRI="Judging_Contracts_Core:Material_Circumstance"/>
        <Class abbreviatedIRI="norm:Normatively_Qualified"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Art1341-1342_Exception"/>
        <Class IRI="#Exception"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Contract_Clause"/>
        <Class abbreviatedIRI="Judging_Contracts_Core:Contractual_Agreement"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Contract_Clause"/>
        <ObjectMaxCardinality cardinality="1">
            <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/>
            <Class IRI="#Oppressive_Status"/>
        </ObjectMaxCardinality>
    </SubClassOf>
    <SubClassOf>
```

```
        <Class IRI="#Efficacious_ViaException"/>
        <Class IRI="#Inefficacious"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Exception"/>
        <Class abbreviatedIRI="Judging_Contracts_Core:Legal_Status"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Full_Contract"/>
        <Class abbreviatedIRI="Judging_Contracts_Core:Contractual_Agreement"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Inefficacious"/>
        <Class abbreviatedIRI="Judging_Contracts_Core:Contractual_Agreement"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Inefficacy"/>
        <Class abbreviatedIRI="Judging_Contracts_Core:Legal_Consequence"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#NonOppressive_Status"/>
        <Class abbreviatedIRI="Judging_Contracts_Core:Legal_Status"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Oppressive_Status"/>
        <Class abbreviatedIRI="Judging_Contracts_Core:Legal_Status"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Relevant_ExArt1341co1"/>
        <Class abbreviatedIRI="norm:Normatively_Qualified"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Relevant_ExArt1341co2"/>
```

```
    <Class abbreviatedIRI="norm:Normatively_Qualified"/>
</SubClassOf>
<SubClassOf>
    <Class IRI="#Relevant_ExArt1342co1"/>
    <Class abbreviatedIRI="Judging_Contracts_Core:Contractual_Agreement"/>
</SubClassOf>
<SubClassOf>
    <Class IRI="#Relevant_ExArt1342co2"/>
    <Class abbreviatedIRI="Judging_Contracts_Core:Contractual_Agreement"/>
</SubClassOf>
<SubClassOf>
    <Class IRI="#Reproducing_LawDisposition"/>
    <Class abbreviatedIRI="norm:Normatively_Qualified"/>
</SubClassOf>
<SubClassOf>
    <Class IRI="#Valid_ExArt1341co1"/>
    <Class abbreviatedIRI="norm:Normatively_Qualified"/>
</SubClassOf>
<SubClassOf>
    <Class IRI="#Valid_ExArt1341co2"/>
    <Class abbreviatedIRI="norm:Normatively_Qualified"/>
</SubClassOf>
<DisjointClasses>
    <Class IRI="#NonOppressive_Status"/>
    <Class IRI="#Oppressive_Status"/>
</DisjointClasses>
<ClassAssertion>
    <Class abbreviatedIRI="Judging_Contracts_Core:Contractual_Agreement"/>
    <NamedIndividual IRI="#AOUMessina_ByteSoftwareHouse_Clause2.020"/>
</ClassAssertion>
<ClassAssertion>
    <Class abbreviatedIRI="Judging_Contracts_Core:Contractual_Agreement"/>
    <NamedIndividual IRI="#AcquedottoPugliese_CC_Clause8"/>
```

```
    &lt;/ClassAssertion&gt;
    &lt;ClassAssertion&gt;
        &lt;Class abbreviatedIRI="Judging_Contracts_Core:Legal_Status"/&gt;
        &lt;NamedIndividual IRI="#AddedToPrecompiled"/&gt;
    &lt;/ClassAssertion&gt;
    &lt;ClassAssertion&gt;
        &lt;Class abbreviatedIRI="Judging_Contracts_Core:Contractual_Agreement"/&gt;
        &lt;NamedIndividual IRI="#AgenziaImmobiliareD_NG&amp;PF_ClauseX"/&gt;
    &lt;/ClassAssertion&gt;
    &lt;ClassAssertion&gt;
        &lt;Class abbreviatedIRI="Judging_Contracts_Core:Contractual_Agreement"/&gt;
        &lt;NamedIndividual IRI="#Alfa_Beta1_Clause8"/&gt;
    &lt;/ClassAssertion&gt;
    &lt;ClassAssertion&gt;
        &lt;Class abbreviatedIRI="Judging_Contracts_Core:Legal_Status"/&gt;
        &lt;NamedIndividual IRI="#Arbitration"/&gt;
    &lt;/ClassAssertion&gt;
    &lt;ClassAssertion&gt;
        &lt;Class IRI="#Oppressive_Status"/&gt;
        &lt;NamedIndividual IRI="#Arbitration"/&gt;
    &lt;/ClassAssertion&gt;
    &lt;ClassAssertion&gt;
        &lt;Class abbreviatedIRI="Judging_Contracts_Core:Legal_Rule"/&gt;
        &lt;NamedIndividual IRI="#Art1219co3cc"/&gt;
    &lt;/ClassAssertion&gt;
    &lt;ClassAssertion&gt;
        &lt;Class abbreviatedIRI="Judging_Contracts_Core:Legal_Rule"/&gt;
        &lt;NamedIndividual IRI="#Art1341co1cc"/&gt;
    &lt;/ClassAssertion&gt;
    &lt;ClassAssertion&gt;
        &lt;Class abbreviatedIRI="Judging_Contracts_Core:Legal_Rule"/&gt;
        &lt;NamedIndividual IRI="#Art1341co2cc"/&gt;
    &lt;/ClassAssertion&gt;
```

```
<ClassAssertion>
    <Class abbreviatedIRI="Judging_Contracts_Core:Legal_Rule"/>
    <NamedIndividual IRI="#Art1342co1cc"/>
</ClassAssertion>
<ClassAssertion>
    <Class abbreviatedIRI="Judging_Contracts_Core:Legal_Rule"/>
    <NamedIndividual IRI="#Art1342co2cc"/>
</ClassAssertion>
<ClassAssertion>
    <Class abbreviatedIRI="Judging_Contracts_Core:Legal_Rule"/>
    <NamedIndividual IRI="#Art1743cc"/>
</ClassAssertion>
<ClassAssertion>
    <Class abbreviatedIRI="Judging_Contracts_Core:Contractual_Agreement"/>
    <NamedIndividual IRI="#AssicurazioniGenerali_Csas_SectionA_Clause2"/>
</ClassAssertion>
<ClassAssertion>
    <Class abbreviatedIRI="Judging_Contracts_Core:Contractual_Agreement"/>
    <NamedIndividual IRI="#AutomobilClubEuropeo_GeCapital_ClauseX"/>
</ClassAssertion>
<ClassAssertion>
    <Class abbreviatedIRI="Judging_Contracts_Core:Contractual_Agreement"/>
    <NamedIndividual IRI="#BC_BancaCreditoCooperativo_Clause6"/>
</ClassAssertion>
<ClassAssertion>
    <Class abbreviatedIRI="Judging_Contracts_Core:Contractual_Agreement"/>
    <NamedIndividual IRI="#BL_Assitalia_Clause2.8"/>
</ClassAssertion>
<ClassAssertion>
    <Class abbreviatedIRI="Judging_Contracts_Core:Contractual_Agreement"/>
    <NamedIndividual IRI="#CG_CarigeAssicurazioni_Clause31"/>
</ClassAssertion>
<ClassAssertion>
```

```
        &lt;Class abbreviatedIRI="Judging_Contracts_Core:Contractual_Agreement"/&gt;
        &lt;NamedIndividual IRI="#CG_IntesaVita_Clause6"/&gt;
    &lt;/ClassAssertion&gt;
    &lt;ClassAssertion&gt;
        &lt;Class abbreviatedIRI="Judging_Contracts_Core:Judicial_Interpretation"/&gt;
        &lt;NamedIndividual IRI="#CdABariII_Int1"/&gt;
    &lt;/ClassAssertion&gt;
    &lt;ClassAssertion&gt;
        &lt;Class abbreviatedIRI="Judging_Contracts_Core:Judicial_Interpretation"/&gt;
        &lt;NamedIndividual IRI="#CdAFirenzeI_Int1"/&gt;
    &lt;/ClassAssertion&gt;
    &lt;ClassAssertion&gt;
        &lt;Class abbreviatedIRI="Judging_Contracts_Core:Judicial_Interpretation"/&gt;
        &lt;NamedIndividual IRI="#CdAMilanoII_Int1"/&gt;
    &lt;/ClassAssertion&gt;
    &lt;ClassAssertion&gt;
        &lt;Class abbreviatedIRI="Judging_Contracts_Core:Judicial_Interpretation"/&gt;
        &lt;NamedIndividual IRI="#CdARomaII_Int1"/&gt;
    &lt;/ClassAssertion&gt;
    &lt;ClassAssertion&gt;
        &lt;Class abbreviatedIRI="Judging_Contracts_Core:Legal_Status"/&gt;
        &lt;NamedIndividual IRI="#CompetenceDerogation"/&gt;
    &lt;/ClassAssertion&gt;
    &lt;ClassAssertion&gt;
        &lt;Class IRI="#Oppressive_Status"/&gt;
        &lt;NamedIndividual IRI="#CompetenceDerogation"/&gt;
    &lt;/ClassAssertion&gt;
    &lt;ClassAssertion&gt;
        &lt;Class abbreviatedIRI="Judging_Contracts_Core:Contractual_Agreement"/&gt;
        &lt;NamedIndividual IRI="#ComunePortoAzzurro_Daneco_Clause21"/&gt;
    &lt;/ClassAssertion&gt;
    &lt;ClassAssertion&gt;
        &lt;Class abbreviatedIRI="Judging_Contracts_Core:Contractual_Agreement"/&gt;
```

```
    &lt;NamedIndividual IRI="#CondominioViaDelGambero_BM_Clause16"/&gt;
&lt;/ClassAssertion&gt;
&lt;ClassAssertion&gt;
    &lt;Class abbreviatedIRI="Judging_Contracts_Core:Contractual_Agreement"/&gt;
    &lt;NamedIndividual IRI="#CondominioViaRasori_Santacroce_ClauseX"/&gt;
&lt;/ClassAssertion&gt;
&lt;ClassAssertion&gt;
    &lt;Class abbreviatedIRI="Judging_Contracts_Core:Legal_Status"/&gt;
    &lt;NamedIndividual IRI="#ContractWithdrawalForBothParties"/&gt;
&lt;/ClassAssertion&gt;
&lt;ClassAssertion&gt;
    &lt;Class IRI="#Oppressive_Status"/&gt;
    &lt;NamedIndividual IRI="#ContractWithdrawalUnilateral"/&gt;
&lt;/ClassAssertion&gt;
&lt;ClassAssertion&gt;
    &lt;Class abbreviatedIRI="Judging_Contracts_Core:Contractual_Agreement"/&gt;
    &lt;NamedIndividual IRI="#EuropaPalaceHotel_SeatPagineGialle_Clause4"/&gt;
&lt;/ClassAssertion&gt;
&lt;ClassAssertion&gt;
    &lt;Class abbreviatedIRI="Judging_Contracts_Core:Legal_Status"/&gt;
    &lt;NamedIndividual IRI="#Exception_Limitation"/&gt;
&lt;/ClassAssertion&gt;
&lt;ClassAssertion&gt;
    &lt;Class IRI="#Oppressive_Status"/&gt;
    &lt;NamedIndividual IRI="#Exception_Limitation"/&gt;
&lt;/ClassAssertion&gt;
&lt;ClassAssertion&gt;
    &lt;Class abbreviatedIRI="Judging_Contracts_Core:Legal_Status"/&gt;
    &lt;NamedIndividual IRI="#ExpertOpinion"/&gt;
&lt;/ClassAssertion&gt;
&lt;ClassAssertion&gt;
    &lt;Class abbreviatedIRI="Judging_Contracts_Core:Legal_Status"/&gt;
    &lt;NamedIndividual IRI="#General"/&gt;
```

```
    </ClassAssertion>
    <ClassAssertion>
        <Class abbreviatedIRI="Judging_Contracts_Core:Contractual_Agreement"/>
        <NamedIndividual IRI="#IBIImmobiliare_AACF_ClauseC"/>
    </ClassAssertion>
    <ClassAssertion>
        <Class abbreviatedIRI="Judging_Contracts_Core:Contractual_Agreement"/>
        <NamedIndividual IRI="#ISPB_SAI_clause_12"/>
    </ClassAssertion>
    <ClassAssertion>
        <Class IRI="#Inefficacy"/>
        <NamedIndividual IRI="#Inefficacy_ExArt1341co1"/>
    </ClassAssertion>
    <ClassAssertion>
        <Class IRI="#Inefficacy"/>
        <NamedIndividual IRI="#Inefficacy_ExArt1341co2"/>
    </ClassAssertion>
    <ClassAssertion>
        <Class IRI="#Inefficacy"/>
        <NamedIndividual IRI="#Inefficacy_ExArt1342co2"/>
    </ClassAssertion>
    <ClassAssertion>
        <Class abbreviatedIRI="Judging_Contracts_Core:Legal_Status"/>
        <NamedIndividual IRI="#International"/>
    </ClassAssertion>
    <ClassAssertion>
        <Class IRI="#Art1341-1342_Exception"/>
        <NamedIndividual IRI="#International"/>
    </ClassAssertion>
    <ClassAssertion>
        <Class abbreviatedIRI="Judging_Contracts_Core:Legal_Status"/>
        <NamedIndividual IRI="#Knowable"/>
    </ClassAssertion>
```

```
<ClassAssertion>
    <Class abbreviatedIRI="Judging_Contracts_Core:Contractual_Agreement"/>
    <NamedIndividual IRI="#LLoyd_FF_Clause*"/>
</ClassAssertion>
<ClassAssertion>
    <Class abbreviatedIRI="Judging_Contracts_Core:Contractual_Agreement"/>
    <NamedIndividual IRI="#LLoyd_FF_Contract"/>
</ClassAssertion>
<ClassAssertion>
    <Class abbreviatedIRI="Judging_Contracts_Core:Legal_Rule"/>
    <NamedIndividual IRI="#Law180_1993Art2co2"/>
</ClassAssertion>
<ClassAssertion>
    <Class abbreviatedIRI="Judging_Contracts_Core:Legal_Status"/>
    <NamedIndividual IRI="#LiabilityLimitation"/>
</ClassAssertion>
<ClassAssertion>
    <Class IRI="#Oppressive_Status"/>
    <NamedIndividual IRI="#LiabilityLimitation"/>
</ClassAssertion>
<ClassAssertion>
    <Class abbreviatedIRI="Judging_Contracts_Core:Legal_Status"/>
    <NamedIndividual IRI="#LimitationTowards3rdParties"/>
</ClassAssertion>
<ClassAssertion>
    <Class IRI="#Oppressive_Status"/>
    <NamedIndividual IRI="#LimitationTowards3rdParties"/>
</ClassAssertion>
<ClassAssertion>
    <Class abbreviatedIRI="Judging_Contracts_Core:Contractual_Agreement"/>
    <NamedIndividual IRI="#ME_LaSorgente_Clause8"/>
</ClassAssertion>
<ClassAssertion>
```

```
    <Class abbreviatedIRI="Judging_Contracts_Core:Contractual_Agreement"/>
    <NamedIndividual IRI="#MeAComputer_Cartasi_ClauseX"/>
</ClassAssertion>
<ClassAssertion>
    <Class abbreviatedIRI="Judging_Contracts_Core:Contractual_Agreement"/>
    <NamedIndividual IRI="#NewEdge_FotoVillage_clause12"/>
</ClassAssertion>
<ClassAssertion>
    <Class IRI="#NonOppressive_Status"/>
    <NamedIndividual IRI="#No_Arbitration"/>
</ClassAssertion>
<ClassAssertion>
    <Class IRI="#NonOppressive_Status"/>
    <NamedIndividual IRI="#No_CompentenceDerogation"/>
</ClassAssertion>
<ClassAssertion>
    <Class IRI="#NonOppressive_Status"/>
    <NamedIndividual IRI="#No_ContractWithdrawal"/>
</ClassAssertion>
<ClassAssertion>
    <Class IRI="#NonOppressive_Status"/>
    <NamedIndividual IRI="#No_ResponsibilityLimitation"/>
</ClassAssertion>
<ClassAssertion>
    <Class abbreviatedIRI="Judging_Contracts_Core:Legal_Status"/>
    <NamedIndividual IRI="#NotSpecificallySigned"/>
</ClassAssertion>
<ClassAssertion>
    <Class abbreviatedIRI="Judging_Contracts_Core:Legal_Status"/>
    <NamedIndividual IRI="#Not_Knowable"/>
</ClassAssertion>
<ClassAssertion>
    <Class abbreviatedIRI="Judging_Contracts_Core:Legal_Status"/>
```

```
    <NamedIndividual IRI="#Not_Unilateral"/>
</ClassAssertion>
<ClassAssertion>
    <Class abbreviatedIRI="Judging_Contracts_Core:Contractual_Agreement"/>
    <NamedIndividual IRI="#NovusWohnbedarfVertriebs_Bspa_Clause3.1"/>
</ClassAssertion>
<ClassAssertion>
    <Class abbreviatedIRI="Judging_Contracts_Core:Legal_Status"/>
    <NamedIndividual IRI="#Precompiled"/>
</ClassAssertion>
<ClassAssertion>
    <Class abbreviatedIRI="Judging_Contracts_Core:Legal_Consequence"/>
    <NamedIndividual IRI="#PrevailsOverPrecompiledClauses"/>
</ClassAssertion>
<ClassAssertion>
    <Class abbreviatedIRI="Judging_Contracts_Core:Legal_Status"/>
    <NamedIndividual IRI="#ReproducingLawDisposition"/>
</ClassAssertion>
<ClassAssertion>
    <Class IRI="#Art1341-1342_Exception"/>
    <NamedIndividual IRI="#ReproducingLawDisposition"/>
</ClassAssertion>
<ClassAssertion>
    <Class abbreviatedIRI="Judging_Contracts_Core:Legal_Status"/>
    <NamedIndividual IRI="#ResponsibilityExtension"/>
</ClassAssertion>
<ClassAssertion>
    <Class abbreviatedIRI="Judging_Contracts_Core:Legal_Status"/>
    <NamedIndividual IRI="#ResponsibilityLimitation"/>
</ClassAssertion>
<ClassAssertion>
    <Class IRI="#Oppressive_Status"/>
    <NamedIndividual IRI="#ResponsibilityLimitation"/>
```

```
</ClassAssertion>
<ClassAssertion>
    <Class abbreviatedIRI="Judging_Contracts_Core:Contractual_Agreement"/>
    <NamedIndividual IRI="#SC_LaPiemonteseAssicurazioni_Clause15f"/>
</ClassAssertion>
<ClassAssertion>
    <Class abbreviatedIRI="Judging_Contracts_Core:Contractual_Agreement"/>
    <NamedIndividual IRI="#STF_KSBClause10.2"/>
</ClassAssertion>
<ClassAssertion>
    <Class abbreviatedIRI="Judging_Contracts_Core:Legal_Status"/>
    <NamedIndividual IRI="#Specific"/>
</ClassAssertion>
<ClassAssertion>
    <Class abbreviatedIRI="Judging_Contracts_Core:Legal_Status"/>
    <NamedIndividual IRI="#SpecificallySigned"/>
</ClassAssertion>
<ClassAssertion>
    <Class abbreviatedIRI="Judging_Contracts_Core:Contractual_Agreement"/>
    <NamedIndividual IRI="#StatutoFIGC_Clause24"/>
</ClassAssertion>
<ClassAssertion>
    <Class abbreviatedIRI="Judging_Contracts_Core:Contractual_Agreement"/>
    <NamedIndividual IRI="#ToroAssicurazioni_ImpresaAutoimpianti_ClauseB"/>
</ClassAssertion>
<ClassAssertion>
    <Class abbreviatedIRI="Judging_Contracts_Core:Judicial_Interpretation"/>
    <NamedIndividual IRI="#TribBariIII_Int1"/>
</ClassAssertion>
<ClassAssertion>
    <Class abbreviatedIRI="Judging_Contracts_Core:Judicial_Interpretation"/>
    <NamedIndividual IRI="#TribComoI_Int1"/>
</ClassAssertion>
```

```
<ClassAssertion>
    <Class abbreviatedIRI="Judging_Contracts_Core:Judicial_Interpretation"/>
    <NamedIndividual IRI="#TribFirenzeIII_Int1"/>
</ClassAssertion>
<ClassAssertion>
    <Class abbreviatedIRI="Judging_Contracts_Core:Judicial_Interpretation"/>
    <NamedIndividual IRI="#TribGenovaI_Int1"/>
</ClassAssertion>
<ClassAssertion>
    <Class abbreviatedIRI="Judging_Contracts_Core:Judicial_Interpretation"/>
    <NamedIndividual IRI="#TribIsernia_int1"/>
</ClassAssertion>
<ClassAssertion>
    <Class abbreviatedIRI="Judging_Contracts_Core:Judicial_Interpretation"/>
    <NamedIndividual IRI="#TribIsernia_int1.2"/>
</ClassAssertion>
<ClassAssertion>
    <Class abbreviatedIRI="Judging_Contracts_Core:Judicial_Interpretation"/>
    <NamedIndividual IRI="#TribModenaII_Int1"/>
</ClassAssertion>
<ClassAssertion>
    <Class abbreviatedIRI="Judging_Contracts_Core:Judicial_Interpretation"/>
    <NamedIndividual IRI="#TribMondovi_Int1"/>
</ClassAssertion>
<ClassAssertion>
    <Class abbreviatedIRI="Judging_Contracts_Core:Judicial_Interpretation"/>
    <NamedIndividual IRI="#TribMonzaI_Int1"/>
</ClassAssertion>
<ClassAssertion>
    <Class abbreviatedIRI="Judging_Contracts_Core:Judicial_Interpretation"/>
    <NamedIndividual IRI="#TribMonzaI_Int2.1"/>
</ClassAssertion>
<ClassAssertion>
```

```
        <Class abbreviatedIRI="Judging_Contracts_Core:Judicial_Interpretation"/>
        <NamedIndividual IRI="#TribNapoliI_Int1"/>
    </ClassAssertion>
    <ClassAssertion>
        <Class abbreviatedIRI="Judging_Contracts_Core:Judicial_Interpretation"/>
        <NamedIndividual IRI="#TribNolaII_Int1"/>
    </ClassAssertion>
    <ClassAssertion>
        <Class abbreviatedIRI="Judging_Contracts_Core:Judicial_Interpretation"/>
        <NamedIndividual IRI="#TribPiacenzaI_Int1"/>
    </ClassAssertion>
    <ClassAssertion>
        <Class abbreviatedIRI="Judging_Contracts_Core:Judicial_Interpretation"/>
        <NamedIndividual IRI="#TribPiacenzaI_Int2.1"/>
    </ClassAssertion>
    <ClassAssertion>
        <Class abbreviatedIRI="Judging_Contracts_Core:Judicial_Interpretation"/>
        <NamedIndividual IRI="#TribRomaIX_Int1"/>
    </ClassAssertion>
    <ClassAssertion>
        <Class abbreviatedIRI="Judging_Contracts_Core:Judicial_Interpretation"/>
        <NamedIndividual IRI="#TribSalernoI_Int1"/>
    </ClassAssertion>
    <ClassAssertion>
        <Class abbreviatedIRI="Judging_Contracts_Core:Judicial_Interpretation"/>
        <NamedIndividual IRI="#TribSavonaI_Int1"/>
    </ClassAssertion>
    <ClassAssertion>
        <Class abbreviatedIRI="Judging_Contracts_Core:Judicial_Interpretation"/>
        <NamedIndividual IRI="#TribTorinoIII_Int1"/>
    </ClassAssertion>
    <ClassAssertion>
        <Class abbreviatedIRI="Judging_Contracts_Core:Judicial_Interpretation"/>
```

```xml
    &lt;NamedIndividual IRI="#TribTorinoIII_Int2.1"/&gt;
&lt;/ClassAssertion&gt;
&lt;ClassAssertion&gt;
    &lt;Class abbreviatedIRI="Judging_Contracts_Core:Judicial_Interpretation"/&gt;
    &lt;NamedIndividual IRI="#TribTorinoI_Int1"/&gt;
&lt;/ClassAssertion&gt;
&lt;ClassAssertion&gt;
    &lt;Class abbreviatedIRI="Judging_Contracts_Core:Judicial_Interpretation"/&gt;
    &lt;NamedIndividual IRI="#TribTorinoVIII_Int1"/&gt;
&lt;/ClassAssertion&gt;
&lt;ClassAssertion&gt;
    &lt;Class abbreviatedIRI="Judging_Contracts_Core:Judicial_Interpretation"/&gt;
    &lt;NamedIndividual IRI="#TribTraniI_Int1"/&gt;
&lt;/ClassAssertion&gt;
&lt;ClassAssertion&gt;
    &lt;Class abbreviatedIRI="Judging_Contracts_Core:Judicial_Interpretation"/&gt;
    &lt;NamedIndividual IRI="#TribVeneziaI_Int1"/&gt;
&lt;/ClassAssertion&gt;
&lt;ClassAssertion&gt;
    &lt;Class abbreviatedIRI="Judging_Contracts_Core:Legal_Status"/&gt;
    &lt;NamedIndividual IRI="#Unilateral"/&gt;
&lt;/ClassAssertion&gt;
&lt;ClassAssertion&gt;
    &lt;Class abbreviatedIRI="Judging_Contracts_Core:Contractual_Agreement"/&gt;
    &lt;NamedIndividual IRI="#VF_BPTR_clauseX"/&gt;
&lt;/ClassAssertion&gt;
&lt;SameIndividual&gt;
    &lt;NamedIndividual IRI="#No_ResponsibilityLimitation"/&gt;
    &lt;NamedIndividual IRI="#ResponsibilityExtension"/&gt;
&lt;/SameIndividual&gt;
&lt;DifferentIndividuals&gt;
    &lt;NamedIndividual IRI="#No_ResponsibilityLimitation"/&gt;
    &lt;NamedIndividual IRI="#ResponsibilityLimitation"/&gt;
```

```
    &lt;/DifferentIndividuals&gt;
    &lt;ObjectPropertyAssertion&gt;
        &lt;ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/&gt;
        &lt;NamedIndividual IRI="#AOUMessina_ByteSoftwareHouse_Clause2.020"/&gt;
        &lt;NamedIndividual IRI="#Arbitration"/&gt;
    &lt;/ObjectPropertyAssertion&gt;
    &lt;ObjectPropertyAssertion&gt;
        &lt;ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/&gt;
        &lt;NamedIndividual IRI="#AOUMessina_ByteSoftwareHouse_Clause2.020"/&gt;
        &lt;NamedIndividual IRI="#Unilateral"/&gt;
    &lt;/ObjectPropertyAssertion&gt;
    &lt;ObjectPropertyAssertion&gt;
        &lt;ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/&gt;
        &lt;NamedIndividual IRI="#AcquedottoPugliese_CC_Clause8"/&gt;
        &lt;NamedIndividual IRI="#Arbitration"/&gt;
    &lt;/ObjectPropertyAssertion&gt;
    &lt;ObjectPropertyAssertion&gt;
        &lt;ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/&gt;
        &lt;NamedIndividual IRI="#Alfa_Beta1_Clause8"/&gt;
        &lt;NamedIndividual IRI="#SpecificallySigned"/&gt;
    &lt;/ObjectPropertyAssertion&gt;
    &lt;ObjectPropertyAssertion&gt;
        &lt;ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/&gt;
        &lt;NamedIndividual IRI="#Art1341co1cc"/&gt;
        &lt;NamedIndividual IRI="#Inefficacy_ExArt1341co1"/&gt;
    &lt;/ObjectPropertyAssertion&gt;
    &lt;ObjectPropertyAssertion&gt;
        &lt;ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/&gt;
        &lt;NamedIndividual IRI="#Art1341co2cc"/&gt;
        &lt;NamedIndividual IRI="#Inefficacy_ExArt1341co2"/&gt;
    &lt;/ObjectPropertyAssertion&gt;
    &lt;ObjectPropertyAssertion&gt;
        &lt;ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/&gt;
```

```
    &lt;NamedIndividual IRI="#Art1342co1cc"/&gt;
    &lt;NamedIndividual IRI="#PrevailsOverPrecompiledClauses"/&gt;
&lt;/ObjectPropertyAssertion&gt;
&lt;ObjectPropertyAssertion&gt;
    &lt;ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/&gt;
    &lt;NamedIndividual IRI="#Art1342co2cc"/&gt;
    &lt;NamedIndividual IRI="#Inefficacy_ExArt1342co2"/&gt;
&lt;/ObjectPropertyAssertion&gt;
&lt;ObjectPropertyAssertion&gt;
    &lt;ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/&gt;
    &lt;NamedIndividual IRI="#AssicurazioniGenerali_Csas_SectionA_Clause2"/&gt;
    &lt;NamedIndividual IRI="#General"/&gt;
&lt;/ObjectPropertyAssertion&gt;
&lt;ObjectPropertyAssertion&gt;
    &lt;ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/&gt;
    &lt;NamedIndividual IRI="#AssicurazioniGenerali_Csas_SectionA_Clause2"/&gt;
    &lt;NamedIndividual IRI="#NotSpecificallySigned"/&gt;
&lt;/ObjectPropertyAssertion&gt;
&lt;ObjectPropertyAssertion&gt;
    &lt;ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/&gt;
    &lt;NamedIndividual IRI="#AssicurazioniGenerali_Csas_SectionA_Clause2"/&gt;
    &lt;NamedIndividual IRI="#Unilateral"/&gt;
&lt;/ObjectPropertyAssertion&gt;
&lt;ObjectPropertyAssertion&gt;
    &lt;ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/&gt;
    &lt;NamedIndividual IRI="#BL_Assitalia_Clause2.8"/&gt;
    &lt;NamedIndividual IRI="#General"/&gt;
&lt;/ObjectPropertyAssertion&gt;
&lt;ObjectPropertyAssertion&gt;
    &lt;ObjectProperty IRI="#assumption"/&gt;
    &lt;NamedIndividual IRI="#BL_Assitalia_Clause2.8"/&gt;
    &lt;NamedIndividual IRI="#Unilateral"/&gt;
&lt;/ObjectPropertyAssertion&gt;
```

```
<ObjectPropertyAssertion>
    <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/>
    <NamedIndividual IRI="#CG_CarigeAssicurazioni_Clause31"/>
    <NamedIndividual IRI="#ExpertOpinion"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
    <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/>
    <NamedIndividual IRI="#CG_IntesaVita_Clause6"/>
    <NamedIndividual IRI="#Unilateral"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
    <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/>
    <NamedIndividual IRI="#CG_IntesaVita_Clause6"/>
    <NamedIndividual IRI="#NotSpecificallySigned"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
    <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/>
    <NamedIndividual IRI="#CG_IntesaVita_Clause6"/>
    <NamedIndividual IRI="#General"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
    <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/>
    <NamedIndividual IRI="#CG_IntesaVita_Clause6"/>
    <NamedIndividual IRI="#Exception_Limitation"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
    <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/>
    <NamedIndividual IRI="#CdABariII_Int1"/>
    <NamedIndividual IRI="#Not_Unilateral"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
    <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:considers"/>
    <NamedIndividual IRI="#CdABariII_Int1"/>
```

```
    <NamedIndividual IRI="#VF_BPTR_clauseX"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
    <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/>
    <NamedIndividual IRI="#CdAFirenzeI_Int1"/>
    <NamedIndividual IRI="#Not_Unilateral"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
    <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/>
    <NamedIndividual IRI="#CdAFirenzeI_Int1"/>
    <NamedIndividual IRI="#Knowable"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
    <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/>
    <NamedIndividual IRI="#CdAFirenzeI_Int1"/>
    <NamedIndividual IRI="#Specific"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
    <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:considers"/>
    <NamedIndividual IRI="#CdAFirenzeI_Int1"/>
    <NamedIndividual IRI="#ComunePortoAzzurro_Daneco_Clause21"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
    <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/>
    <NamedIndividual IRI="#CdAMilanoII_Int1"/>
    <NamedIndividual IRI="#Not_Unilateral"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
    <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:considers"/>
    <NamedIndividual IRI="#CdAMilanoII_Int1"/>
    <NamedIndividual IRI="#CondominioViaRasori_Santacroce_ClauseX"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
```

```xml
        <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/>
        <NamedIndividual IRI="#CdARomaII_Int1"/>
        <NamedIndividual IRI="#ResponsibilityExtension"/>
    </ObjectPropertyAssertion>
    <ObjectPropertyAssertion>
        <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:considers"/>
        <NamedIndividual IRI="#CdARomaII_Int1"/>
        <NamedIndividual IRI="#ToroAssicurazioni_ImpresaAutoimpianti_ClauseB"/>
    </ObjectPropertyAssertion>
    <ObjectPropertyAssertion>
        <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/>
        <NamedIndividual IRI="#ComunePortoAzzurro_Daneco_Clause21"/>
        <NamedIndividual IRI="#Arbitration"/>
    </ObjectPropertyAssertion>
    <ObjectPropertyAssertion>
        <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/>
        <NamedIndividual IRI="#CondominioViaDelGambero_BM_Clause16"/>
        <NamedIndividual IRI="#Arbitration"/>
    </ObjectPropertyAssertion>
    <ObjectPropertyAssertion>
        <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/>
        <NamedIndividual IRI="#CondominioViaRasori_Santacroce_ClauseX"/>
        <NamedIndividual IRI="#NotSpecificallySigned"/>
    </ObjectPropertyAssertion>
    <ObjectPropertyAssertion>
        <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/>
        <NamedIndividual IRI="#CondominioViaRasori_Santacroce_ClauseX"/>
        <NamedIndividual IRI="#Arbitration"/>
    </ObjectPropertyAssertion>
    <ObjectPropertyAssertion>
        <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/>
        <NamedIndividual IRI="#CondominioViaRasori_Santacroce_ClauseX"/>
        <NamedIndividual IRI="#General"/>
```

```
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
    <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/>
    <NamedIndividual IRI="#EuropaPalaceHotel_SeatPagineGialle_Clause4"/>
    <NamedIndividual IRI="#General"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
    <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/>
    <NamedIndividual IRI="#IBIImmobiliare_AACF_ClauseC"/>
    <NamedIndividual IRI="#SpecificallySigned"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
    <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/>
    <NamedIndividual IRI="#ISPB_SAI_clause_12"/>
    <NamedIndividual IRI="#Unilateral"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
    <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/>
    <NamedIndividual IRI="#LLoyd_FF_Clause*"/>
    <NamedIndividual IRI="#AddedToPrecompiled"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
    <ObjectProperty abbreviatedIRI="mereology:contained_in"/>
    <NamedIndividual IRI="#LLoyd_FF_Clause*"/>
    <NamedIndividual IRI="#LLoyd_FF_Contract"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
    <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/>
    <NamedIndividual IRI="#LLoyd_FF_Contract"/>
    <NamedIndividual IRI="#Precompiled"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
    <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/>
```

```
        &lt;NamedIndividual IRI="#ME_LaSorgente_Clause8"/&gt;
        &lt;NamedIndividual IRI="#General"/&gt;
    &lt;/ObjectPropertyAssertion&gt;
    &lt;ObjectPropertyAssertion&gt;
        &lt;ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/&gt;
        &lt;NamedIndividual IRI="#ME_LaSorgente_Clause8"/&gt;
        &lt;NamedIndividual IRI="#CompetenceDerogation"/&gt;
    &lt;/ObjectPropertyAssertion&gt;
    &lt;ObjectPropertyAssertion&gt;
        &lt;ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/&gt;
        &lt;NamedIndividual IRI="#ME_LaSorgente_Clause8"/&gt;
        &lt;NamedIndividual IRI="#Unilateral"/&gt;
    &lt;/ObjectPropertyAssertion&gt;
    &lt;ObjectPropertyAssertion&gt;
        &lt;ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/&gt;
        &lt;NamedIndividual IRI="#MeAComputer_Cartasi_ClauseX"/&gt;
        &lt;NamedIndividual IRI="#Precompiled"/&gt;
    &lt;/ObjectPropertyAssertion&gt;
    &lt;ObjectPropertyAssertion&gt;
        &lt;ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/&gt;
        &lt;NamedIndividual IRI="#MeAComputer_Cartasi_ClauseX"/&gt;
        &lt;NamedIndividual IRI="#NotSpecificallySigned"/&gt;
    &lt;/ObjectPropertyAssertion&gt;
    &lt;ObjectPropertyAssertion&gt;
        &lt;ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/&gt;
        &lt;NamedIndividual IRI="#MeAComputer_Cartasi_ClauseX"/&gt;
        &lt;NamedIndividual IRI="#Unilateral"/&gt;
    &lt;/ObjectPropertyAssertion&gt;
    &lt;ObjectPropertyAssertion&gt;
        &lt;ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/&gt;
        &lt;NamedIndividual IRI="#MeAComputer_Cartasi_ClauseX"/&gt;
        &lt;NamedIndividual IRI="#ContractWithdrawalForBothParties"/&gt;
    &lt;/ObjectPropertyAssertion&gt;
```

```
&lt;ObjectPropertyAssertion&gt;
    &lt;ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/&gt;
    &lt;NamedIndividual IRI="#NewEdge_FotoVillage_clause12"/&gt;
    &lt;NamedIndividual IRI="#CompetenceDerogation"/&gt;
&lt;/ObjectPropertyAssertion&gt;
&lt;ObjectPropertyAssertion&gt;
    &lt;ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/&gt;
    &lt;NamedIndividual IRI="#NewEdge_FotoVillage_clause12"/&gt;
    &lt;NamedIndividual IRI="#General"/&gt;
&lt;/ObjectPropertyAssertion&gt;
&lt;ObjectPropertyAssertion&gt;
    &lt;ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/&gt;
    &lt;NamedIndividual IRI="#NewEdge_FotoVillage_clause12"/&gt;
    &lt;NamedIndividual IRI="#Unilateral"/&gt;
&lt;/ObjectPropertyAssertion&gt;
&lt;ObjectPropertyAssertion&gt;
    &lt;ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/&gt;
    &lt;NamedIndividual IRI="#NovusWohnbedarfVertriebs_Bspa_Clause3.1"/&gt;
    &lt;NamedIndividual IRI="#LimitationTowards3rdParties"/&gt;
&lt;/ObjectPropertyAssertion&gt;
&lt;ObjectPropertyAssertion&gt;
    &lt;ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/&gt;
    &lt;NamedIndividual IRI="#SC_LaPiemonteseAssicurazioni_Clause15f"/&gt;
    &lt;NamedIndividual IRI="#General"/&gt;
&lt;/ObjectPropertyAssertion&gt;
&lt;ObjectPropertyAssertion&gt;
    &lt;ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/&gt;
    &lt;NamedIndividual IRI="#SC_LaPiemonteseAssicurazioni_Clause15f"/&gt;
    &lt;NamedIndividual IRI="#Unilateral"/&gt;
&lt;/ObjectPropertyAssertion&gt;
&lt;ObjectPropertyAssertion&gt;
    &lt;ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/&gt;
    &lt;NamedIndividual IRI="#SC_LaPiemonteseAssicurazioni_Clause15f"/&gt;
```

```
        <NamedIndividual IRI="#NotSpecificallySigned"/>
    </ObjectPropertyAssertion>
    <ObjectPropertyAssertion>
        <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/>
        <NamedIndividual IRI="#STF_KSBClause10.2"/>
        <NamedIndividual IRI="#CompetenceDerogation"/>
    </ObjectPropertyAssertion>
    <ObjectPropertyAssertion>
        <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/>
        <NamedIndividual IRI="#STF_KSBClause10.2"/>
        <NamedIndividual IRI="#General"/>
    </ObjectPropertyAssertion>
    <ObjectPropertyAssertion>
        <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/>
        <NamedIndividual IRI="#StatutoFIGC_Clause24"/>
        <NamedIndividual IRI="#Arbitration"/>
    </ObjectPropertyAssertion>
    <ObjectPropertyAssertion>
        <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/>
        <NamedIndividual IRI="#TribBariIII_Int1"/>
        <NamedIndividual IRI="#NotSpecificallySigned"/>
    </ObjectPropertyAssertion>
    <ObjectPropertyAssertion>
        <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:considers"/>
        <NamedIndividual IRI="#TribBariIII_Int1"/>
        <NamedIndividual IRI="#AcquedottoPugliese_CC_Clause8"/>
    </ObjectPropertyAssertion>
    <ObjectPropertyAssertion>
        <ObjectProperty IRI="#assumption"/>
        <NamedIndividual IRI="#TribBariIII_Int1"/>
        <NamedIndividual IRI="#General"/>
    </ObjectPropertyAssertion>
    <ObjectPropertyAssertion>
```

```xml
        <ObjectProperty IRI="#assumption"/>
        <NamedIndividual IRI="#TribBariIII_Int1"/>
        <NamedIndividual IRI="#Unilateral"/>
    </ObjectPropertyAssertion>
    <ObjectPropertyAssertion>
        <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/>
        <NamedIndividual IRI="#TribComoI_Int1"/>
        <NamedIndividual IRI="#NotSpecificallySigned"/>
    </ObjectPropertyAssertion>
    <ObjectPropertyAssertion>
        <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:considers"/>
        <NamedIndividual IRI="#TribComoI_Int1"/>
        <NamedIndividual IRI="#ME_LaSorgente_Clause8"/>
    </ObjectPropertyAssertion>
    <ObjectPropertyAssertion>
        <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/>
        <NamedIndividual IRI="#TribFirenzeIII_Int1"/>
        <NamedIndividual IRI="#NotSpecificallySigned"/>
    </ObjectPropertyAssertion>
    <ObjectPropertyAssertion>
        <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:considers"/>
        <NamedIndividual IRI="#TribFirenzeIII_Int1"/>
        <NamedIndividual IRI="#LLoyd_FF_Clause*"/>
    </ObjectPropertyAssertion>
    <ObjectPropertyAssertion>
        <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:considers"/>
        <NamedIndividual IRI="#TribGenovaI_Int1"/>
        <NamedIndividual IRI="#IBIImmobiliare_AACF_ClauseC"/>
    </ObjectPropertyAssertion>
    <ObjectPropertyAssertion>
        <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/>
        <NamedIndividual IRI="#TribIsernia_int1"/>
        <NamedIndividual IRI="#No_CompentenceDerogation"/>
```

```
    </ObjectPropertyAssertion>
    <ObjectPropertyAssertion>
        <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:considers"/>
        <NamedIndividual IRI="#TribIsernia_int1"/>
        <NamedIndividual IRI="#ExpertOpinion"/>
    </ObjectPropertyAssertion>
    <ObjectPropertyAssertion>
        <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:considers"/>
        <NamedIndividual IRI="#TribIsernia_int1.2"/>
        <NamedIndividual IRI="#CG_CarigeAssicurazioni_Clause31"/>
    </ObjectPropertyAssertion>
    <ObjectPropertyAssertion>
        <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/>
        <NamedIndividual IRI="#TribModenaII_Int1"/>
        <NamedIndividual IRI="#Not_Unilateral"/>
    </ObjectPropertyAssertion>
    <ObjectPropertyAssertion>
        <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:considers"/>
        <NamedIndividual IRI="#TribModenaII_Int1"/>
        <NamedIndividual IRI="#CondominioViaDelGambero_BM_Clause16"/>
    </ObjectPropertyAssertion>
    <ObjectPropertyAssertion>
        <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/>
        <NamedIndividual IRI="#TribMondovi_Int1"/>
        <NamedIndividual IRI="#SpecificallySigned"/>
    </ObjectPropertyAssertion>
    <ObjectPropertyAssertion>
        <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:considers"/>
        <NamedIndividual IRI="#TribMondovi_Int1"/>
        <NamedIndividual IRI="#AutomobilClubEuropeo_GeCapital_ClauseX"/>
    </ObjectPropertyAssertion>
    <ObjectPropertyAssertion>
        <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/>
```

```
    <NamedIndividual IRI="#TribMonzaI_Int1"/>
    <NamedIndividual IRI="#Knowable"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
    <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:considers"/>
    <NamedIndividual IRI="#TribMonzaI_Int1"/>
    <NamedIndividual IRI="#Alfa_Beta1_Clause8"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
    <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/>
    <NamedIndividual IRI="#TribMonzaI_Int2.1"/>
    <NamedIndividual IRI="#Art1743cc"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
    <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:considers"/>
    <NamedIndividual IRI="#TribMonzaI_Int2.1"/>
    <NamedIndividual IRI="#NovusWohnbedarfVertriebs_Bspa_Clause3.1"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
    <ObjectProperty IRI="#exception"/>
    <NamedIndividual IRI="#TribMonzaI_Int2.1"/>
    <NamedIndividual IRI="#ReproducingLawDisposition"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
    <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/>
    <NamedIndividual IRI="#TribNapoliI_Int1"/>
    <NamedIndividual IRI="#Arbitration"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
    <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:considers"/>
    <NamedIndividual IRI="#TribNapoliI_Int1"/>
    <NamedIndividual IRI="#BL_Assitalia_Clause2.8"/>
</ObjectPropertyAssertion>
```

```
&lt;ObjectPropertyAssertion&gt;
    &lt;ObjectProperty IRI="#assumption"/&gt;
    &lt;NamedIndividual IRI="#TribNapoliI_Int1"/&gt;
    &lt;NamedIndividual IRI="#NotSpecificallySigned"/&gt;
&lt;/ObjectPropertyAssertion&gt;
&lt;ObjectPropertyAssertion&gt;
    &lt;ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/&gt;
    &lt;NamedIndividual IRI="#TribNolaII_Int1"/&gt;
    &lt;NamedIndividual IRI="#No_ContractWithdrawal"/&gt;
&lt;/ObjectPropertyAssertion&gt;
&lt;ObjectPropertyAssertion&gt;
    &lt;ObjectProperty abbreviatedIRI="Judging_Contracts_Core:considers"/&gt;
    &lt;NamedIndividual IRI="#TribNolaII_Int1"/&gt;
    &lt;NamedIndividual IRI="#MeAComputer_Cartasi_ClauseX"/&gt;
&lt;/ObjectPropertyAssertion&gt;
&lt;ObjectPropertyAssertion&gt;
    &lt;ObjectProperty abbreviatedIRI="Judging_Contracts_Core:considers"/&gt;
    &lt;NamedIndividual IRI="#TribNolaII_Int1"/&gt;
    &lt;NamedIndividual IRI="#ContractWithdrawalForBothParties"/&gt;
&lt;/ObjectPropertyAssertion&gt;
&lt;ObjectPropertyAssertion&gt;
    &lt;ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/&gt;
    &lt;NamedIndividual IRI="#TribPiacenzaI_Int1"/&gt;
    &lt;NamedIndividual IRI="#NotSpecificallySigned"/&gt;
&lt;/ObjectPropertyAssertion&gt;
&lt;ObjectPropertyAssertion&gt;
    &lt;ObjectProperty abbreviatedIRI="Judging_Contracts_Core:considers"/&gt;
    &lt;NamedIndividual IRI="#TribPiacenzaI_Int1"/&gt;
    &lt;NamedIndividual IRI="#NewEdge_FotoVillage_clause12"/&gt;
&lt;/ObjectPropertyAssertion&gt;
&lt;ObjectPropertyAssertion&gt;
    &lt;ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/&gt;
    &lt;NamedIndividual IRI="#TribPiacenzaI_Int2.1"/&gt;
```

```
    &lt;NamedIndividual IRI="#Knowable"/&gt;
&lt;/ObjectPropertyAssertion&gt;
&lt;ObjectPropertyAssertion&gt;
    &lt;ObjectProperty abbreviatedIRI="Judging_Contracts_Core:considers"/&gt;
    &lt;NamedIndividual IRI="#TribPiacenzaI_Int2.1"/&gt;
    &lt;NamedIndividual IRI="#STF_KSBClause10.2"/&gt;
&lt;/ObjectPropertyAssertion&gt;
&lt;ObjectPropertyAssertion&gt;
    &lt;ObjectProperty IRI="#exception"/&gt;
    &lt;NamedIndividual IRI="#TribPiacenzaI_Int2.1"/&gt;
    &lt;NamedIndividual IRI="#International"/&gt;
&lt;/ObjectPropertyAssertion&gt;
&lt;ObjectPropertyAssertion&gt;
    &lt;ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/&gt;
    &lt;NamedIndividual IRI="#TribRomaIX_Int1"/&gt;
    &lt;NamedIndividual IRI="#Art1219co3cc"/&gt;
&lt;/ObjectPropertyAssertion&gt;
&lt;ObjectPropertyAssertion&gt;
    &lt;ObjectProperty abbreviatedIRI="Judging_Contracts_Core:considers"/&gt;
    &lt;NamedIndividual IRI="#TribRomaIX_Int1"/&gt;
    &lt;NamedIndividual IRI="#CG_IntesaVita_Clause6"/&gt;
&lt;/ObjectPropertyAssertion&gt;
&lt;ObjectPropertyAssertion&gt;
    &lt;ObjectProperty IRI="#exception"/&gt;
    &lt;NamedIndividual IRI="#TribRomaIX_Int1"/&gt;
    &lt;NamedIndividual IRI="#ReproducingLawDisposition"/&gt;
&lt;/ObjectPropertyAssertion&gt;
&lt;ObjectPropertyAssertion&gt;
    &lt;ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/&gt;
    &lt;NamedIndividual IRI="#TribSalernoI_Int1"/&gt;
    &lt;NamedIndividual IRI="#Law180_1993Art2co2"/&gt;
&lt;/ObjectPropertyAssertion&gt;
&lt;ObjectPropertyAssertion&gt;
```

```
        &lt;ObjectProperty abbreviatedIRI="Judging_Contracts_Core:considers"/&gt;
        &lt;NamedIndividual IRI="#TribSalernoI_Int1"/&gt;
        &lt;NamedIndividual IRI="#StatutoFIGC_Clause24"/&gt;
    &lt;/ObjectPropertyAssertion&gt;
    &lt;ObjectPropertyAssertion&gt;
        &lt;ObjectProperty IRI="#exception"/&gt;
        &lt;NamedIndividual IRI="#TribSalernoI_Int1"/&gt;
        &lt;NamedIndividual IRI="#ReproducingLawDisposition"/&gt;
    &lt;/ObjectPropertyAssertion&gt;
    &lt;ObjectPropertyAssertion&gt;
        &lt;ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/&gt;
        &lt;NamedIndividual IRI="#TribSavonaI_Int1"/&gt;
        &lt;NamedIndividual IRI="#SpecificallySigned"/&gt;
    &lt;/ObjectPropertyAssertion&gt;
    &lt;ObjectPropertyAssertion&gt;
        &lt;ObjectProperty abbreviatedIRI="Judging_Contracts_Core:considers"/&gt;
        &lt;NamedIndividual IRI="#TribSavonaI_Int1"/&gt;
        &lt;NamedIndividual IRI="#BC_BancaCreditoCooperativo_Clause6"/&gt;
    &lt;/ObjectPropertyAssertion&gt;
    &lt;ObjectPropertyAssertion&gt;
        &lt;ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/&gt;
        &lt;NamedIndividual IRI="#TribTorinoIII_Int1"/&gt;
        &lt;NamedIndividual IRI="#No_ResponsibilityLimitation"/&gt;
    &lt;/ObjectPropertyAssertion&gt;
    &lt;ObjectPropertyAssertion&gt;
        &lt;ObjectProperty abbreviatedIRI="Judging_Contracts_Core:considers"/&gt;
        &lt;NamedIndividual IRI="#TribTorinoIII_Int1"/&gt;
        &lt;NamedIndividual IRI="#SC_LaPiemonteseAssicurazioni_Clause15f"/&gt;
    &lt;/ObjectPropertyAssertion&gt;
    &lt;ObjectPropertyAssertion&gt;
        &lt;ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/&gt;
        &lt;NamedIndividual IRI="#TribTorinoIII_Int2.1"/&gt;
        &lt;NamedIndividual IRI="#SpecificallySigned"/&gt;
```

```xml
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
    <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:considers"/>
    <NamedIndividual IRI="#TribTorinoIII_Int2.1"/>
    <NamedIndividual IRI="#EuropaPalaceHotel_SeatPagineGialle_Clause4"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
    <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/>
    <NamedIndividual IRI="#TribTorinoI_Int1"/>
    <NamedIndividual IRI="#General"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
    <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/>
    <NamedIndividual IRI="#TribTorinoI_Int1"/>
    <NamedIndividual IRI="#LiabilityLimitation"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
    <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/>
    <NamedIndividual IRI="#TribTorinoI_Int1"/>
    <NamedIndividual IRI="#NotSpecificallySigned"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
    <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:considers"/>
    <NamedIndividual IRI="#TribTorinoI_Int1"/>
    <NamedIndividual IRI="#ISPB_SAI_clause_12"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
    <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/>
    <NamedIndividual IRI="#TribTorinoVIII_Int1"/>
    <NamedIndividual IRI="#Not_Knowable"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
    <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/>
```

```
        <NamedIndividual IRI="#TribTorinoVIII_Int1"/>
        <NamedIndividual IRI="#NotSpecificallySigned"/>
    </ObjectPropertyAssertion>
    <ObjectPropertyAssertion>
        <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/>
        <NamedIndividual IRI="#TribTorinoVIII_Int1"/>
        <NamedIndividual IRI="#Unilateral"/>
    </ObjectPropertyAssertion>
    <ObjectPropertyAssertion>
        <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/>
        <NamedIndividual IRI="#TribTorinoVIII_Int1"/>
        <NamedIndividual IRI="#General"/>
    </ObjectPropertyAssertion>
    <ObjectPropertyAssertion>
        <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:considers"/>
        <NamedIndividual IRI="#TribTorinoVIII_Int1"/>
        <NamedIndividual IRI="#AOUMessina_ByteSoftwareHouse_Clause2.020"/>
    </ObjectPropertyAssertion>
    <ObjectPropertyAssertion>
        <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/>
        <NamedIndividual IRI="#TribTraniI_Int1"/>
        <NamedIndividual IRI="#SpecificallySigned"/>
    </ObjectPropertyAssertion>
    <ObjectPropertyAssertion>
        <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:considers"/>
        <NamedIndividual IRI="#TribTraniI_Int1"/>
        <NamedIndividual IRI="#AgenziaImmobiliareD_NG&amp;PF_ClauseX"/>
    </ObjectPropertyAssertion>
    <ObjectPropertyAssertion>
        <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/>
        <NamedIndividual IRI="#TribVeneziaI_Int1"/>
        <NamedIndividual IRI="#LiabilityLimitation"/>
    </ObjectPropertyAssertion>
```

```xml
<ObjectPropertyAssertion>
    <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:considers"/>
    <NamedIndividual IRI="#TribVeneziaI_Int1"/>
    <NamedIndividual IRI="#AssicurazioniGenerali_Csas_SectionA_Clause2"/>
</ObjectPropertyAssertion>
<SubObjectPropertyOf>
    <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applied_by"/>
    <ObjectProperty abbreviatedIRI="expression:qualified_by"/>
</SubObjectPropertyOf>
<SubObjectPropertyOf>
    <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/>
    <ObjectProperty abbreviatedIRI="expression:qualifies"/>
</SubObjectPropertyOf>
<SubObjectPropertyOf>
    <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:considered_by"/>
    <ObjectProperty abbreviatedIRI="expression:qualified_by"/>
</SubObjectPropertyOf>
<SubObjectPropertyOf>
    <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:considers"/>
    <ObjectProperty abbreviatedIRI="expression:qualifies"/>
</SubObjectPropertyOf>
<SubObjectPropertyOf>
    <ObjectProperty IRI="#assumption"/>
    <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/>
</SubObjectPropertyOf>
<SubObjectPropertyOf>
    <ObjectProperty IRI="#exception"/>
    <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/>
</SubObjectPropertyOf>
<SubObjectPropertyOf>
    <ObjectProperty abbreviatedIRI="expression:evaluated_by"/>
    <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:considered_by"/>
</SubObjectPropertyOf>
```

```
<SubObjectPropertyOf>
    <ObjectProperty abbreviatedIRI="expression:evaluates"/>
    <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:considers"/>
</SubObjectPropertyOf>
<SubObjectPropertyOf>
    <ObjectProperty abbreviatedIRI="norm:allowed_by"/>
    <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:considered_by"/>
</SubObjectPropertyOf>
<SubObjectPropertyOf>
    <ObjectProperty abbreviatedIRI="norm:allows"/>
    <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:considers"/>
</SubObjectPropertyOf>
<SubObjectPropertyOf>
    <ObjectProperty abbreviatedIRI="norm:disallowed_by"/>
    <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:considered_by"/>
</SubObjectPropertyOf>
<SubObjectPropertyOf>
    <ObjectProperty abbreviatedIRI="norm:disallows"/>
    <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:considers"/>
</SubObjectPropertyOf>
<InverseObjectProperties>
    <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applied_by"/>
    <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/>
</InverseObjectProperties>
<InverseObjectProperties>
    <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:considers"/>
    <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:considered_by"/>
</InverseObjectProperties>
<SubObjectPropertyOf>
    <ObjectPropertyChain>
        <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:considered_by"/>
        <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:applies"/>
    </ObjectPropertyChain>
```

```
        <ObjectProperty abbreviatedIRI="Judging_Contracts_Core:judged_as"/>

    </SubObjectPropertyOf>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>

        <IRI>#AOUMessina_ByteSoftwareHouse_Clause2.020</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">clause 2.020 of contract between Byte Software House spa and Azienda Ospedaliera Universitaria di messina policlinico Gaetano Martino</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:label"/>

        <IRI>#AOUMessina_ByteSoftwareHouse_Clause2.020</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">AOU Messina / Byte Software House Clause 2.020</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>

        <IRI>#AcquedottoPugliese_CC_Clause8</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">clause 8 of the contract between Acquedotto Pugliese spa and C. C.</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:label"/>

        <IRI>#AcquedottoPugliese_CC_Clause8</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Acquedotto Pugliese / CC Clause 8</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>

        <IRI>#AddedToPrecompiled</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Clause has been manually added to a precompiled contract</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:label"/>

        <IRI>#AddedToPrecompiled</IRI>
```

```
        <Literal datatypeIRI="&rdf;PlainLiteral">Added to precompiled</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>

        <IRI>#AgenziaImmobiliareD_NG&amp;PF_ClauseX</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">clause x of the contract between Agenzia Immobiliare D. di M. D. and N. G., P. F.</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:label"/>

        <IRI>#AgenziaImmobiliareD_NG&amp;PF_ClauseX</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Agenzia Immobiliare D / NG e PF Clause &#39;x&#39;</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>

        <IRI>#Alfa_Beta1_Clause8</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Clause 8 of contract between unknown parties</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:label"/>

        <IRI>#Alfa_Beta1_Clause8</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Alfa1 / Beta1 clause 8</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>

        <IRI>#Arbitration</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Clause concerns arbitration</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:label"/>

        <IRI>#Arbitration</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Arbitration</Literal>
```

```
    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:label"/>

        <IRI>#Art1219co3cc</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Article 1219 comma 3 Italian Civil Code</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>

        <IRI>#Art1341-1342_Exception</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Contract clauses applying any of these statuses are pulled out of the scope of application of article 1341 and 1342 civil code.</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:label"/>

        <IRI>#Art1341-1342_Exception</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Exception to articles 1341 and 1342 civil code</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>

        <IRI>#Art1341co1cc</IRI>

        <Literal xml:lang="it" datatypeIRI="&rdf;PlainLiteral">Articolo 1341 comma 1 Codice Civile: &#39;Le condizioni generali di contratto predisposte da uno dei contraenti sono efficaci nei confronti dell’altro, se al momento della conclusione del contratto questi le ha conosciute o avrebbe dovuto conoscerle usando l’ordinaria diligenza.&#39;</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:label"/>

        <IRI>#Art1341co1cc</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Article 1341 comma 1 Italian Civil Code</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
```

```
        <IRI>#Art1341co2cc</IRI>

        <Literal xml:lang="it" datatypeIRI="&rdf;PlainLiteral">Articolo 1341 comma 2 Codice Civile: &#39;In ogni caso non hanno effetto, se non sono specificamente approvate per iscritto, le condizioni che stabiliscono, a favore di colui che le ha predisposte, limitazioni di responsabilità, facoltà di recedere dal contratto o di sospenderne l&#39;esecuzione, ovvero sanciscono a carico dell&#39;altro contraente decadenze, limitazioni alla facoltà di opporre eccezioni, restrizioni alla libertà contrattuale nei rapporti coi terzi, tacita proroga o rinnovazione del contratto, clausole compromissorie o deroghe alla competenza dell&#39;autorità giudiziaria.&#39;</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:label"/>

        <IRI>#Art1341co2cc</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Article 1341 comma 2 Italian Civil Code</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>

        <IRI>#Art1342co1cc</IRI>

        <Literal xml:lang="it" datatypeIRI="&rdf;PlainLiteral">Articolo 1342 comma 1 Codice Civile: &#39;Nei contratti conclusi mediante la sottoscrizione di moduli o formulari, predisposti per disciplinare in maniera uniforme determinati rapporti contrattuali, le clausole aggiunte al modulo o al formulario prevalgono su quelle del modulo o del formulario qualora siano incompatibili con esse, anche se queste ultime non sono state cancellate.&#39;</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:label"/>

        <IRI>#Art1342co1cc</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Article 1342 comma 1 Italian Civil Code</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>

        <IRI>#Art1342co2cc</IRI>

        <Literal xml:lang="it" datatypeIRI="&rdf;PlainLiteral">Articolo 1342 comma 2 Codice Civile: &quot;Si osserva inoltre la disposizione del secondo comma dell&#39;articolo precedente.&quot;</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:label"/>
```

```
        <IRI>#Art1342co2cc</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Article 1342 comma 2 Italian Civil
Code</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>

        <IRI>#Art1743cc</IRI>

        <Literal xml:lang="it" datatypeIRI="&rdf;PlainLiteral">Articolo 1743 Codice
Civile: &quot;Il preponente non può valersi contemporaneamente di più agenti nella
stessa zona e per lo stesso ramo di attività, né l&#39;agente può assumere
l&#39;incarico di trattare nella stessa zona e per lo stesso ramo gli affari di più
imprese in concorrenza tra loro.&quot;</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:label"/>

        <IRI>#Art1743cc</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Article 1743 Italian Civil
Code</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>

        <IRI>#AssicurazioniGenerali_Csas_SectionA_Clause2</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Clause 2 of contract between
Assicurazione Generali Spa and C. sas.</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:label"/>

        <IRI>#AssicurazioniGenerali_Csas_SectionA_Clause2</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Assicurazioni Generali / C. sas Clause
2</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>

        <IRI>#AutomobilClubEuropeo_GeCapital_ClauseX</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Clause unknown of contract between
Automobil Club Europeo srl and Ge Capital Servizi Finanziari spa</Literal>

    </AnnotationAssertion>
```

```
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:label"/>
        <IRI>#AutomobilClubEuropeo_GeCapital_ClauseX</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">AutomobilClubEuropeo / GeCapital Clause
&#39;x&#39;</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#BC_BancaCreditoCooperativo_Clause6</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Clause 6 of the contract between B. C.
and Banca Credito Cooperativo di Alba, Langhe e Roero scrl</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:label"/>
        <IRI>#BC_BancaCreditoCooperativo_Clause6</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">B.C. / Banca di Credito Cooperativo
Clause 6</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#BL_Assitalia_Clause2.8</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Clause 2.8 of contract between B.L. and
Assitalia spa</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:label"/>
        <IRI>#BL_Assitalia_Clause2.8</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">B.L. / Assitalia clause 2.8</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#CG_CarigeAssicurazioni_Clause31</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Clause 31 of contract between C. G. and
Carige Assicurazioni spa</Literal>
    </AnnotationAssertion>
```

```
    &lt;AnnotationAssertion&gt;
        &lt;AnnotationProperty abbreviatedIRI="rdfs:label"/&gt;
        &lt;IRI&gt;#CG_CarigeAssicurazioni_Clause31&lt;/IRI&gt;
        &lt;Literal datatypeIRI="&rdf;PlainLiteral"&gt;C. G. / CarigeAssicurazioni_Clause
31&lt;/Literal&gt;
    &lt;/AnnotationAssertion&gt;
    &lt;AnnotationAssertion&gt;
        &lt;AnnotationProperty abbreviatedIRI="rdfs:comment"/&gt;
        &lt;IRI&gt;#CG_IntesaVita_Clause6&lt;/IRI&gt;
        &lt;Literal datatypeIRI="&rdf;PlainLiteral"&gt;Clause 6 of contract between C. G. and
Intesa Vita spa&lt;/Literal&gt;
    &lt;/AnnotationAssertion&gt;
    &lt;AnnotationAssertion&gt;
        &lt;AnnotationProperty abbreviatedIRI="rdfs:label"/&gt;
        &lt;IRI&gt;#CG_IntesaVita_Clause6&lt;/IRI&gt;
        &lt;Literal datatypeIRI="&rdf;PlainLiteral"&gt;C. G. / Intesavita clause 6&lt;/Literal&gt;
    &lt;/AnnotationAssertion&gt;
    &lt;AnnotationAssertion&gt;
        &lt;AnnotationProperty abbreviatedIRI="rdfs:comment"/&gt;
        &lt;IRI&gt;#CdABariII_Int1&lt;/IRI&gt;
        &lt;Literal datatypeIRI="&rdf;PlainLiteral"&gt;Interpretation by the Tribunal of Bari,
3rd section in decision n. 461 of May 25th, 2004&lt;/Literal&gt;
    &lt;/AnnotationAssertion&gt;
    &lt;AnnotationAssertion&gt;
        &lt;AnnotationProperty abbreviatedIRI="rdfs:label"/&gt;
        &lt;IRI&gt;#CdABariII_Int1&lt;/IRI&gt;
        &lt;Literal datatypeIRI="&rdf;PlainLiteral"&gt;Court of Appeal of Bari Interpretation
1&lt;/Literal&gt;
    &lt;/AnnotationAssertion&gt;
    &lt;AnnotationAssertion&gt;
        &lt;AnnotationProperty abbreviatedIRI="rdfs:comment"/&gt;
        &lt;IRI&gt;#CdAFirenzeI_Int1&lt;/IRI&gt;
        &lt;Literal datatypeIRI="&rdf;PlainLiteral"&gt;Interpretation by the Court of Appeal
of Firenze, 1st section, in decision of December 10th, 2004&lt;/Literal&gt;
    &lt;/AnnotationAssertion&gt;
```

```
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:label"/>
        <IRI>#CdAFirenzeI_Int1</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Court of Appeal of Firenze interpretation 1</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#CdAMilanoII_Int1</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">interpretation by the Court of Appeal of Milano 2nd section in decision n. 1154 of April 24th, 2007</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:label"/>
        <IRI>#CdAMilanoII_Int1</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Court of Appeal of Milano interpretation 1</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#CdARomaII_Int1</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Interpretation by the Court of Appeal of Rome 2nd section in sentence n. 628 of February 14th, 2008.</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:label"/>
        <IRI>#CdARomaII_Int1</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Court of Appeal of Roma interpretation 1</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#CompetenceDerogation</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Clause concerns a derogation to the judicial competence</Literal>
    </AnnotationAssertion>
```

```
    &lt;AnnotationAssertion&gt;
        &lt;AnnotationProperty abbreviatedIRI="rdfs:label"/&gt;
        &lt;IRI&gt;#CompetenceDerogation&lt;/IRI&gt;
        &lt;Literal datatypeIRI="&rdf;PlainLiteral"&gt;Competence derogation&lt;/Literal&gt;
    &lt;/AnnotationAssertion&gt;
    &lt;AnnotationAssertion&gt;
        &lt;AnnotationProperty abbreviatedIRI="rdfs:comment"/&gt;
        &lt;IRI&gt;#ComunePortoAzzurro_Daneco_Clause21&lt;/IRI&gt;
        &lt;Literal datatypeIRI="&rdf;PlainLiteral"&gt;Clause 21 of contract between Comune di Porto Azzurro and Daneco Gestione Impianti spa&lt;/Literal&gt;
    &lt;/AnnotationAssertion&gt;
    &lt;AnnotationAssertion&gt;
        &lt;AnnotationProperty abbreviatedIRI="rdfs:label"/&gt;
        &lt;IRI&gt;#ComunePortoAzzurro_Daneco_Clause21&lt;/IRI&gt;
        &lt;Literal datatypeIRI="&rdf;PlainLiteral"&gt;Comune Porto Azzurro / Daneco Clause 21&lt;/Literal&gt;
    &lt;/AnnotationAssertion&gt;
    &lt;AnnotationAssertion&gt;
        &lt;AnnotationProperty abbreviatedIRI="rdfs:comment"/&gt;
        &lt;IRI&gt;#CondominioViaDelGambero_BM_Clause16&lt;/IRI&gt;
        &lt;Literal datatypeIRI="&rdf;PlainLiteral"&gt;Clause 16 of contract between Condominio di via del Gambero n. 40 and B. M.&lt;/Literal&gt;
    &lt;/AnnotationAssertion&gt;
    &lt;AnnotationAssertion&gt;
        &lt;AnnotationProperty abbreviatedIRI="rdfs:label"/&gt;
        &lt;IRI&gt;#CondominioViaDelGambero_BM_Clause16&lt;/IRI&gt;
        &lt;Literal datatypeIRI="&rdf;PlainLiteral"&gt;Condominio Via del Gambero / B. M. clause 16&lt;/Literal&gt;
    &lt;/AnnotationAssertion&gt;
    &lt;AnnotationAssertion&gt;
        &lt;AnnotationProperty abbreviatedIRI="rdfs:comment"/&gt;
        &lt;IRI&gt;#CondominioViaRasori_Santacroce_ClauseX&lt;/IRI&gt;
        &lt;Literal datatypeIRI="&rdf;PlainLiteral"&gt;clause unknown of contract between Condominio di via Rasori n. 9 and Impresa Edile Santacroce sas&lt;/Literal&gt;
    &lt;/AnnotationAssertion&gt;
```

```
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:label"/>
        <IRI>#CondominioViaRasori_Santacroce_ClauseX</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Condominio Via Rasori_
Santacroce_Clause x</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#ContractWithdrawalForBothParties</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Clause allow particular conditions of withdrawal for both parties</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:label"/>
        <IRI>#ContractWithdrawalForBothParties</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Contract withdrawal for both parties</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#Contract_Clause</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">A part of a contract</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:label"/>
        <IRI>#Contract_Clause</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Contract Clause</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#Efficacious_ViaException</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Contractual agreements classified under this are relevant under some laws which disposes inefficacy, but are nevertheless efficacious because they fall into the scope of some exception.</Literal>
    </AnnotationAssertion>
```

```
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:label"/>
        <IRI>#Efficacious_ViaException</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Efficacious through Exception</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#EuropaPalaceHotel_SeatPagineGialle_Clause4</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">clause 4 of contract between Europa Palace Hotel spa and Seat Pagine Gialle spa</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:label"/>
        <IRI>#EuropaPalaceHotel_SeatPagineGialle_Clause4</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Europa Palace Hotel / Seat Pagine Gialle clause 4</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#Exception</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">An exceptional status: it pulls the expression it is applied to out of some norm´s scope of application.</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:label"/>
        <IRI>#Exception</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Exception</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#Exception_Limitation</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Clause concerns a limitation to the exceptions that can be opposed to the proponant</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
```

```
        <AnnotationProperty abbreviatedIRI="rdfs:label"/>

        <IRI>#Exception_Limitation</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Limitation to exceptions</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>

        <IRI>#ExpertOpinion</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Clause concerns an expert
opinion</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:label"/>

        <IRI>#ExpertOpinion</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Expert Opinion</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>

        <IRI>#Full_Contract</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">The expression representing the whole
contract. Can be further identified by contract clauses.</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:label"/>

        <IRI>#Full_Contract</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Full Contract</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>

        <IRI>#General</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Clause is general (a general condition
is common to a series of contracts)</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:label"/>
```

```
        <IRI>#General</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">General</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>

        <IRI>#IBIImmobiliare_AACF_ClauseC</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Clause C of contract between IBI Immobiliare Italiana srl and A.A., C. F.</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:label"/>

        <IRI>#IBIImmobiliare_AACF_ClauseC</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">IBI Immobiliare / A.A. and C. F. clause C</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>

        <IRI>#ISPB_SAI_clause_12</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Clause 12 of contract between Istituto Sorveglianza Provinciale Bergamasco spa and SAI spa.</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:label"/>

        <IRI>#ISPB_SAI_clause_12</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">ISPB / SAI clause 12</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>

        <IRI>#Inefficacious</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">An agreement (contract or clause) to which the legal consequence ´inefficacy´ has been applied.</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:label"/>

        <IRI>#Inefficacious</IRI>
```

```
        <Literal datatypeIRI="&rdf;PlainLiteral">Inefficacious Agreement</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>

        <IRI>#Inefficacy</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">The legal consequence of inefficacy:
the contractual agreement (or other legal expression) has no effects.</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:label"/>

        <IRI>#Inefficacy</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Inefficacy</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>

        <IRI>#Inefficacy_ExArt1341co1</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Clause is inefficacious under article
1341 comma 1 Civil Code</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:label"/>

        <IRI>#Inefficacy_ExArt1341co1</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Inefficacy under article 1341 comma 1
Civil Code</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>

        <IRI>#Inefficacy_ExArt1341co2</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Clause is inefficacious under article
1341 comma 2 Civil Code</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:label"/>

        <IRI>#Inefficacy_ExArt1341co2</IRI>
```

```xml
        <Literal datatypeIRI="&rdf;PlainLiteral">Inefficacy under article 1341 comma 2 Civil Code</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#Inefficacy_ExArt1342co2</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Clause is inefficacious under article 1342 comma 2 Civil Code</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:label"/>
        <IRI>#Inefficacy_ExArt1342co2</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Inefficacy under article 1342 comma 2 Civil Code</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#International</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">The contract is international: its parties are of different countries.</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:label"/>
        <IRI>#International</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">International</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#Knowable</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Clause was knowable by the other party at the time of the conclusion of the contract by using ordinary diligence</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:label"/>
        <IRI>#Knowable</IRI>
```

```
        <Literal datatypeIRI="&rdf;PlainLiteral">Knowable</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#LLoyd_FF_Clause*</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Clause &#39;*&#39; between Lloyd Nazionale Assicurazioni spa and F.F. Consulenze e Assicurazioni sas</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:label"/>
        <IRI>#LLoyd_FF_Clause*</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Lloyd&#39;s / F.F. clause *</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#LLoyd_FF_Contract</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Contract between Lloyd Nazionale Assicurazioni spa and F.F. Consulenze Assicurazioni sas</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:label"/>
        <IRI>#LLoyd_FF_Contract</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Lloyd&#39;s / F.F. contract</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:label"/>
        <IRI>#Law180_1993Art2co2</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Article 2 comma 2 Law 180/1993</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#LiabilityLimitation</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Clause concerns a limitation to the liability of the proponant</Literal>
```

```
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:label"/>
        <IRI>#LiabilityLimitation</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Liability limitation</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#LimitationTowards3rdParties</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Clause concerns a limitation to
contractual freedom towards third parties</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:label"/>
        <IRI>#LimitationTowards3rdParties</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Limitation towards third
parties</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#ME_LaSorgente_Clause8</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Clause 8 of contract between M. E. and
La Sorgente sas</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:label"/>
        <IRI>#ME_LaSorgente_Clause8</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">M.E. / La Sorgente clause 8</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#MeAComputer_Cartasi_ClauseX</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">clause x of contract between M &amp; A
Computer sas and Cartasì spa</Literal>
    </AnnotationAssertion>
```

```
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:label"/>
        <IRI>#MeAComputer_Cartasi_ClauseX</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">M &amp; A Computer / Cartasì Clause &#39;x&#39;</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#NewEdge_FotoVillage_clause12</IRI>
        <Literal xml:lang="it" datatypeIRI="&rdf;PlainLiteral">Clause 12 of contract between New Edge di D. F. and Foto Village s.a.s. </Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:label"/>
        <IRI>#NewEdge_FotoVillage_clause12</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">New Edge / Foto Village clause 12</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#NonOppressive_Status</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">non oppressive status (under the consumer law conception of oppressiveness)</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:label"/>
        <IRI>#NonOppressive_Status</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Not oppressive </Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#NotSpecificallySigned</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Clause has not been specifically signed</Literal>
    </AnnotationAssertion>
```

```
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:label"/>
        <IRI>#NotSpecificallySigned</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Not specifically signed</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#Not_Knowable</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Clause was not knowable by the other party at the time of the conclusion of the contract, even using ordinary diligence</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:label"/>
        <IRI>#Not_Knowable</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Non knowable</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#Not_Unilateral</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Clause was not unilaterally predisponed by the proponant</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:label"/>
        <IRI>#Not_Unilateral</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Not unilateral</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#NovusWohnbedarfVertriebs_Bspa_Clause3.1</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Clause 3.1 of contract between Novus Wohnbedarf Vertriebs gmbh and B. spa</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
```

```xml
        &lt;AnnotationProperty abbreviatedIRI="rdfs:label"/&gt;

        &lt;IRI&gt;#NovusWohnbedarfVertriebs_Bspa_Clause3.1&lt;/IRI&gt;

        &lt;Literal datatypeIRI="&rdf;PlainLiteral"&gt;Novus Wothnbedarf Vertriebs / B. spa clause 3.1&lt;/Literal&gt;

    &lt;/AnnotationAssertion&gt;

    &lt;AnnotationAssertion&gt;

        &lt;AnnotationProperty abbreviatedIRI="rdfs:comment"/&gt;

        &lt;IRI&gt;#Oppressive_Status&lt;/IRI&gt;

        &lt;Literal datatypeIRI="&rdf;PlainLiteral"&gt;oppressive status (under the consumer law conception of oppressiveness)&lt;/Literal&gt;

    &lt;/AnnotationAssertion&gt;

    &lt;AnnotationAssertion&gt;

        &lt;AnnotationProperty abbreviatedIRI="rdfs:label"/&gt;

        &lt;IRI&gt;#Oppressive_Status&lt;/IRI&gt;

        &lt;Literal datatypeIRI="&rdf;PlainLiteral"&gt;Oppressive&lt;/Literal&gt;

    &lt;/AnnotationAssertion&gt;

    &lt;AnnotationAssertion&gt;

        &lt;AnnotationProperty abbreviatedIRI="rdfs:comment"/&gt;

        &lt;IRI&gt;#Precompiled&lt;/IRI&gt;

        &lt;Literal datatypeIRI="&rdf;PlainLiteral"&gt;Contract was precompiled by the proponant&lt;/Literal&gt;

    &lt;/AnnotationAssertion&gt;

    &lt;AnnotationAssertion&gt;

        &lt;AnnotationProperty abbreviatedIRI="rdfs:label"/&gt;

        &lt;IRI&gt;#Precompiled&lt;/IRI&gt;

        &lt;Literal datatypeIRI="&rdf;PlainLiteral"&gt;Precompiled&lt;/Literal&gt;

    &lt;/AnnotationAssertion&gt;

    &lt;AnnotationAssertion&gt;

        &lt;AnnotationProperty abbreviatedIRI="rdfs:comment"/&gt;

        &lt;IRI&gt;#PrevailsOverPrecompiledClauses&lt;/IRI&gt;

        &lt;Literal datatypeIRI="&rdf;PlainLiteral"&gt;An added clause prevails over precompiled clauses of the contract following Article 1342 comma 1 Civil Code&lt;/Literal&gt;

    &lt;/AnnotationAssertion&gt;

    &lt;AnnotationAssertion&gt;

        &lt;AnnotationProperty abbreviatedIRI="rdfs:label"/&gt;
```

```
        <IRI>#PrevailsOverPrecompiledClauses</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Prevails over precompiled clauses</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>

        <IRI>#Relevant_ExArt1341co1</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">To be relevan uder comma 1 of article 1341 civil code, a clause must be general, unilaterally written by one of the parties, and not knowable by the other party at the time of the conclusion of the contract by using ordinary diligence.</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:label"/>

        <IRI>#Relevant_ExArt1341co1</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Relevant under article 1341 comma 1 civil code</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>

        <IRI>#Relevant_ExArt1341co2</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">To be relevant under article 1341 comma 2 of civil code a clause must be oppressive, general, unilaterally written by one of the parties and not specifically signed by the other party.</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:label"/>

        <IRI>#Relevant_ExArt1341co2</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Relevant under article 1341 comma 2 civil code</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>

        <IRI>#Relevant_ExArt1342co1</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">To be relevant under comma 1 of article 1342 civil code a clause must constitute an addition to a precompiled contract.</Literal>

    </AnnotationAssertion>
```

```
    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:label"/>

        <IRI>#Relevant_ExArt1342co1</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Relevant under article 1342 comma 1
civil code</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>

        <IRI>#Relevant_ExArt1342co2</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">To be relevant under comma 2 of article
1342 civil code a clause must be relevant under comma 1 and not specifically signed by
the party which did not prepare the precompiled contract.</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:label"/>

        <IRI>#Relevant_ExArt1342co2</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Relevant under article 1342 comma 2
civil code</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>

        <IRI>#ReproducingLawDisposition</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Clause reproduces a law
disposition</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:label"/>

        <IRI>#ReproducingLawDisposition</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Reproducing law disposition</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>

        <IRI>#Reproducing_LawDisposition</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">a clause whose content corresponds to
that of a law disposition in force.</Literal>

    </AnnotationAssertion>
```

```
    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:label"/>

        <IRI>#Reproducing_LawDisposition</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Reproducing law disposition</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>

        <IRI>#ResponsibilityExtension</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Clause concerns an extention of the responsibility of the acceptant</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:label"/>

        <IRI>#ResponsibilityExtension</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Extension of responsibility</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>

        <IRI>#ResponsibilityLimitation</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Clause concerns a limitation to the responsibility of the proponant</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:label"/>

        <IRI>#ResponsibilityLimitation</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Limitation of responsibility</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>

        <IRI>#SC_LaPiemonteseAssicurazioni_Clause15f</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">clause 15f of contract between S. C. and la Piemontese Assicurazioni spa</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>
```

```
        <AnnotationProperty abbreviatedIRI="rdfs:label"/>
        <IRI>#SC_LaPiemonteseAssicurazioni_Clause15f</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">S.C. / La Piemontese Assicurazioni Clause 15f</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#STF_KSBClause10.2</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">clause 10.2 of contract between STF spa and KSB ag</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:label"/>
        <IRI>#STF_KSBClause10.2</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">STF / KSB clause 10.2</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#Specific</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Clause is specific (not general)</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:label"/>
        <IRI>#Specific</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Specific</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#SpecificallySigned</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Clause was specifically signed</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:label"/>
```

```
        <IRI>#SpecificallySigned</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Specifically signed</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#StatutoFIGC_Clause24</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Clause 24 of the Statute of the FIGC,
Federazione Italiana Giuoco Calcio</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:label"/>
        <IRI>#StatutoFIGC_Clause24</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Statuto F. I. G. C. clause 24</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#ToroAssicurazioni_ImpresaAutoimpianti_ClauseB</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Clause B of contract between Impresa
Autoimpianti di L. A. and Toro Assicurazioni spa</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:label"/>
        <IRI>#ToroAssicurazioni_ImpresaAutoimpianti_ClauseB</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Toro Assicurazioni / Impresa
Autoimpianti clause B</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#TribBariIII_Int1</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Interpretation by the Tribunal of Bari
3rd section in decision n. 87 of January 15th, 2009</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:label"/>
        <IRI>#TribBariIII_Int1</IRI>
```

```
        <Literal datatypeIRI="&rdf;PlainLiteral">Tribunal of Bari interpretation 1</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>

        <IRI>#TribComoI_Int1</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Interpretation by the Tribunal of Como in decision of May 20th, 2004</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:label"/>

        <IRI>#TribComoI_Int1</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Tribunal of Como interpretation 1</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>

        <IRI>#TribFirenzeIII_Int1</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">interpretation by the Tribunal of Firenze 3rd section in decision of November 7th, 2006</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:label"/>

        <IRI>#TribFirenzeIII_Int1</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Tribunal of Firenze interpretation 1</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>

        <IRI>#TribGenovaI_Int1</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Interpretation by the Tribunal of Genova in decision of February 19th, 2003</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:label"/>

        <IRI>#TribGenovaI_Int1</IRI>
```

```
        <Literal datatypeIRI="&rdf;PlainLiteral">Tribunal of Genova interpretation
1</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>

        <IRI>#TribIsernia_int1</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Interpretation of the Tribunal of
Isernia in decision n. 65 of January 20th, 2010</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:label"/>

        <IRI>#TribIsernia_int1</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Tribunal of Isernia interpretation
1</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>

        <IRI>#TribIsernia_int1.2</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Interpretation of the Tribunal of
Isernia in decision n. 65 of January 20th, 2010</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:label"/>

        <IRI>#TribIsernia_int1.2</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Tribunal of Isernia interpretation
2</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>

        <IRI>#TribModenaII_Int1</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">interpretation by the Tribunal of
Modena 2nd section in decision of November 2nd, 2005</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:label"/>

        <IRI>#TribModenaII_Int1</IRI>
```

```
        <Literal datatypeIRI="&rdf;PlainLiteral">Tribunal of Modena interpretation 1</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>

        <IRI>#TribMondovi_Int1</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">interpretation of the Tribunal of Mondovì in decision of November 14th, 2006</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:label"/>

        <IRI>#TribMondovi_Int1</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Tribunal of Mondovì interpretation 1</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>

        <IRI>#TribMonzaI_Int1</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Interpretation by the Tribunal of Monza in decision of January 21st, 2003</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:label"/>

        <IRI>#TribMonzaI_Int1</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Tribunal of Monza interpretation 1</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>

        <IRI>#TribMonzaI_Int2.1</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">interpretation by the Tribunal of Monza in decision of November 9th, 2006</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:label"/>

        <IRI>#TribMonzaI_Int2.1</IRI>
```

```
        <Literal datatypeIRI="&rdf;PlainLiteral">Tribunal of Monza interpretation 2</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>

        <IRI>#TribNapoliI_Int1</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Interpretation by the Tribunal of Napoli in decision of January 13th, 2005</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:label"/>

        <IRI>#TribNapoliI_Int1</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Tribunal of Napoli interpretation 1</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>

        <IRI>#TribNolaII_Int1</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Interpretation by the Tribunal of Nola 2nd section in decision of may 20th, 2008</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:label"/>

        <IRI>#TribNolaII_Int1</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Tribunal of Nola interpretation 1</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>

        <IRI>#TribPiacenzaI_Int1</IRI>

        <Literal xml:lang="it" datatypeIRI="&rdf;PlainLiteral">Interpretation by the Tribunal of Piacenza on sentence 507 of July 6th, 2009</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:label"/>

        <IRI>#TribPiacenzaI_Int1</IRI>
```

```
        <Literal datatypeIRI="&rdf;PlainLiteral">Tribunal of Piacenza interpretation 1</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>

        <IRI>#TribPiacenzaI_Int2.1</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Interpretation by the Tribunal of Piacenza in decision n. 599 of September 21st, 2009</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:label"/>

        <IRI>#TribPiacenzaI_Int2.1</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Tribunal of Piacenza interpretation 2</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>

        <IRI>#TribRomaIX_Int1</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Interpretation by the Tribunal of Rome 9th section in decision n. 147 of January 4th, 2008</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:label"/>

        <IRI>#TribRomaIX_Int1</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Tribunal of Roma interpretation 1</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>

        <IRI>#TribSalernoI_Int1</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Interpretation by the Tribunal of Salerno 1st section in decision of October 31st, 2006</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:label"/>

        <IRI>#TribSalernoI_Int1</IRI>
```

```
        <Literal datatypeIRI="&rdf;PlainLiteral">Tribunal of Salerno interpretation 1</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>

        <IRI>#TribSavonaI_Int1</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Interpretation by the Tribunal of Savona in decision of July 24th, 2005</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:label"/>

        <IRI>#TribSavonaI_Int1</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Tribunal of Savona interpretation 1</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>

        <IRI>#TribTorinoIII_Int1</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Interpretation by the Tribunal of Torino 3rd section in decision of October 1st, 2008</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:label"/>

        <IRI>#TribTorinoIII_Int1</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Tribunal of Torino interpretation 1</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>

        <IRI>#TribTorinoIII_Int2.1</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Interpretation by the Tribunal of Torino 3rd section in decision n. 7785 of november 25th, 2008</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:label"/>

        <IRI>#TribTorinoIII_Int2.1</IRI>
```

```
        <Literal datatypeIRI="&rdf;PlainLiteral">Tribunal of Torino interpretation 2</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#TribTorinoI_Int1</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Interpretation by Tribunal of Torino in sentence January 13th, 2003</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:label"/>
        <IRI>#TribTorinoI_Int1</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Tribunal of Torino interpretation 3</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#TribTorinoVIII_Int1</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">interpretation by the Tribunal of Torino, 8th section, in decision of November 26th, 2005</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:label"/>
        <IRI>#TribTorinoVIII_Int1</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Tribunal of Torino interpretation 4</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#TribTraniI_Int1</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Interpretation by the Tribunal of Trani in decision n. 23 of January 23rd, 2009</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:label"/>
        <IRI>#TribTraniI_Int1</IRI>
```

```
        <Literal datatypeIRI="&rdf;PlainLiteral">Tribunal of Trani interpretation
1</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#TribVeneziaI_Int1</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Interpretation by the Tribunal of
Venezia, in sentence of July 11th, 2002</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:label"/>
        <IRI>#TribVeneziaI_Int1</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Tribunal of Venezia interpretation
1</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#Unilateral</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Clause was unilaterally
predisponed</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:label"/>
        <IRI>#Unilateral</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Unilateral</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <IRI>#VF_BPTR_clauseX</IRI>
        <Literal datatypeIRI="&rdf;PlainLiteral">Clause unknown of the contract between
V. F. and F. B. srl</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:label"/>
        <IRI>#VF_BPTR_clauseX</IRI>
```

```
        <Literal datatypeIRI="&rdf;PlainLiteral">V. F. / B. P. &amp; T. R. clause &#39;x&#39;</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>

        <IRI>#Valid_ExArt1341co1</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">A clause which is valid under article 1341 comma 1 civil code (not inefficacious)</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:label"/>

        <IRI>#Valid_ExArt1341co1</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Valid under article 1341 comma 1 civil code</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>

        <IRI>#Valid_ExArt1341co2</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">A clause which is valid (and therefore not inefficacious) under article 1341 comma 2 civil code.</Literal>

    </AnnotationAssertion>

    <AnnotationAssertion>

        <AnnotationProperty abbreviatedIRI="rdfs:label"/>

        <IRI>#Valid_ExArt1341co2</IRI>

        <Literal datatypeIRI="&rdf;PlainLiteral">Valid under article 1341 comma 2 civil code</Literal>

    </AnnotationAssertion>

</Ontology>

<!-- Generated by the OWL API (version 3.2.3.1824)
```

a.2. The rules

a.2.1. in LKIF language

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE rb [

        <!ENTITY oss "http://www.semanticweb.org/ontologies/2011/8/prova1.owl#" >

]>

<?xml-stylesheet type="text/css" href="../../../../../../schemas/LKIF.css"?>

<?oxygen RNGSchema="../../../../../../schemas/LKIF.rnc" type="compact"?>

<lkif>

  <theory id="art1341">

    <imports>

    </imports>

<rules>



<rule id="LAW_Art1341co1">

               <head>

         <s pred="Relevant_ExArt1341co1"><v>C1</v> falls under the discipline of Article 1341 comma 1 of Civil Code </s>

<s pred="&oss;considered_by"><v>C1</v> falls under the discipline of <i value="&oss;Art1341co1cc">Article 1341 comma 1 of Civil Code </i></s>

        </head>

<body>

<s pred="&oss;applies"><v>C1</v> applies <i value="&oss;General"> general status</i> </s>

<s pred="&oss;applies"><v>C1</v> applies <i value="&oss;Unilateral"> unilateral status</i> </s>

<not><s pred="&oss;applies"><v>C1</v> not applies <i value="&oss;Knowledgeable"> knowledgeable before contract sign</i> </s></not>

</body>

      </rule>



<rule id="LAW_Art1341co2">

               <head>

         <s pred="Relevant_ExArt1341co2"><v>C1</v> falls under the discipline of Article 1341 comma 2 of Civil Code </s>

<s pred="&oss;considered_by"><v>C1</v> falls under the discipline of <i value="&oss;Art1341co2cc">Article 1341 comma 2 of Civil Code </i></s>
```

```
</head>

<body>

<s pred="&oss;applies"><v>C1</v> applies <v>S1</v> </s>

<s pred="&oss;Oppressive_Status"><v>S1</v> is an oppressive status</s>

<s pred="&oss;applies"><v>C1</v> applies <i value="&oss;General"> general status</i>
</s>

<s pred="&oss;applies"><v>C1</v> applies <i value="&oss;Unilateral"> unilateral
status</i> </s>

<not><s pred="&oss;applies"><v>C1</v> not applies <i value="&oss;SpecificallySigned">
specifically signed</i> </s></not>

</body>

</rule>



<rule id="LAWCONS_Inefficacy rule">

<head>

<s pred="&oss;Inefficacious"><v>C1</v> Is inefficacious: has no effects </s>

</head>

<body>

<and><or><s pred="Relevant_ExArt1341co1"><v>C1</v> falls under the discipline of Article
1341 comma 1 of Civil Code </s>

<s pred="Relevant_ExArt1341co2"><v>C1</v> falls under the discipline of Article 1341
comma 2 of Civil Code </s>

<s pred="Relevant_ExArt1342co2"><v>C1</v> falls under the discipline of Article 1342
comma 2 of Civil Code </s></or>

<not exception="true"><s pred="&oss;applies"><v>C1</v> applies <i
value="&oss;ReproducingLawDisposition"> a law disposition</i> </s></not>

<not exception="true"><s pred="&oss;applies"><v>C1</v> applies <i
value="&oss;International"> an international agreement</i> </s></not></and>

</body>

</rule>

<rule id="TEC_JudicialSubsumption">

<head>

<s pred="&oss;applies"><v>C1</v> applies status <v>S2</v> </s>

</head>

<body>

<s pred="&oss;judged_as"><v>C1</v> is judged as <v>S2</v> </s>

</body>
```

```
      </rule>


<rule id="NEGINST_NotSpecificallySigned">

 <head>

<not><s pred="&oss;applies"><v>C1</v> applies <i value="&oss;SpecificallySigned"> specifically signed status </i></s></not>

</head>

<body>

<s pred="&oss;applies"><v>C1</v> applies <i value="&oss;NotSpecificallySigned"> specifically signed status </i></s>

</body>

      </rule>

<rule id="NEGINST_NotKnowledgeable">

 <head>

<not><s pred="&oss;applies"><v>C1</v> applies <i value="&oss;Knowledgeable"> specifically signed status </i></s></not>

</head>

<body>

<s pred="&oss;applies"><v>C1</v> applies <i value="&oss;Not_Knowledgeable"> specifically signed status </i></s>

</body>

      </rule>

<rule id="NEGINST_NotGeneral">

 <head>

<not><s pred="&oss;applies"><v>C1</v> applies <i value="&oss;General"> general status </i></s></not>

</head>

<body>

<s pred="&oss;applies"><v>C1</v> applies <i value="&oss;Specific"> specific status </i></s>

</body>

      </rule>

<rule id="NEGINST_NotUnilateral">

 <head>

<not><s pred="&oss;applies"><v>C1</v> applies <i value="&oss;Unilateral"> unilateral status </i></s></not>
```

```
</head>

<body>

<s pred="&oss;applies"><v>C1</v> applies <i value="&oss;Not_Unilateral"> not unilateral status </i></s>

</body>

      </rule>

<rule id="LAWEXCL_Art1341co1">

 <head>

<not><s pred="Relevant_ExArt1341co1"><v>C1</v> doesn't fall under the discipline of Article 1341 comma 1 of Civil Code </s></not>

</head>

<body>

<or><s pred="&oss;applies"><v>C1</v> applies <i value="&oss;SpecificallySigned"> specifically signed status </i></s>

<s pred="&oss;applies"><v>C1</v> applies <i value="&oss;Specific"> specific status </i></s>

<not><s pred="&oss;applies"><v>C1</v> applies <i value="&oss;Unilateral"> not unilateral status </i></s></not>

<s pred="&oss;applies"><v>C1</v> applies <i value="&oss;Knowledgeable"> knowledgeable status </i></s></or>

</body>

      </rule>

<rule id="LAWEXCL_Art1341co2">

 <head>

<not><s pred="Relevant_ExArt1341co2"><v>C1</v> falls under the discipline of Article 1341 comma 2 of Civil Code </s></not>

</head>

<body>

<or><s pred="&oss;applies"><v>C1</v> applies <i value="&oss;SpecificallySigned"> specifically signed status </i></s>

<s pred="&oss;applies"><v>C1</v> applies <i value="&oss;Specific"> specific status </i></s>

<not><s pred="&oss;applies"><v>C1</v> applies <i value="&oss;Unilateral"> not unilateral status </i></s></not></or>

</body>

      </rule>

</rules>

  </theory>
```

```
<theory id="art1342">

    <imports>

    </imports>

<rules>

<rule id="LAW_Art1342co1">

               <head>

         <s pred="Relevant_ExArt1342co1"><v>C1</v> falls under the discipline of Article 1342 comma 1 of Civil Code </s>

<s pred="&oss;considered_by"><v>C1</v> falls under the discipline of <i value="&oss;Art1342co1cc">Article 1342 comma 1 of Civil Code </i></s>

                     <s pred="&oss;normatively_striclty_worse"><v>C1</v> is applied instead of<v>C2</v></s>

        </head>

<body>

<s pred="&oss;contained_in"><v>C1</v> is contained in <v>A1</v> </s>

      <s pred="&oss;contained_in"><v>C2</v> is contained in <v>A1</v> </s>

<s pred="&oss;applies"><v>A1</v> is <i value="&oss;Precompiled">precompiled </i></s>

<s pred="&oss;applies"><v>C1</v> <i value="&oss;AddedToPrecompiled">is added to precompiled </i></s>

<not exception="true"> <s pred="&oss;applies"><v>C2</v>applies <i value="&oss;AddedToPrecompiled">is added to precompiled </i></s></not>

</body>

      </rule>

<rule id="LAW_Art1342co2">

               <head>

         <s pred="Relevant_ExArt1342co2"><v>C1</v> falls under the discipline of Article 1342 comma 2 of Civil Code </s>

<s pred="&oss;considered_by"><v>C1</v> falls under the discipline of <i value="&oss;Art1342co2cc">Article 1342 comma 2 of Civil Code </i></s>

        </head>

<body>

<s pred="&oss;contained_in"><v>C1</v> is contained in <v>A1</v> </s>

<s pred="&oss;applies"><v>A1</v> is <i value="&oss;Precompiled">precompiled </i></s>

<s pred="&oss;applies"><v>C1</v> <i value="&oss;AddedToPrecompiled">is added to precompiled </i></s>

<not><s pred="&oss;applies"><v>C1</v> <i value="&oss;SpecificallySigned">is specifically signed </i></s></not>
```

```
&lt;/body&gt;

      &lt;/rule&gt;

      &lt;rule id="NEGINST_NotPrecompiled"&gt;

            &lt;head&gt;

                  &lt;not&gt;<s       pred="&oss;applies">&lt;v&gt;A1&lt;/v&gt;       is       not       <i
value="&oss;Precompiled">a precompiled contract</i></s>&lt;/not&gt;

            &lt;/head&gt;

            &lt;body&gt;

                              <s        pred="&oss;applies">&lt;v&gt;A1&lt;/v&gt;        is        <i
value="&oss;NotPrecompiled">a not precompiled contract</i></s>

&lt;/body&gt;

      &lt;/rule&gt;

&lt;/rules&gt;

  &lt;/theory&gt;

&lt;theory id="CompromissoryClauseRules"&gt;

        &lt;imports&gt;

       &lt;/imports&gt;

        &lt;rules&gt;

            &lt;rule id="JINT_NoCompromissory1"&gt;

                &lt;head&gt;

                    &lt;not&gt;<s        pred="&oss;applies">&lt;v&gt;C1&lt;/v&gt;        is        not<i
value="&oss;ArbitrationAgreement">an arbitration clause</i></s>&lt;/not&gt;

                &lt;/head&gt;

                &lt;body&gt;

                    <s        pred="&oss;hasfactor">&lt;v&gt;C1&lt;/v&gt;        has        factor<i
value="&oss;BindingSurvey">excludes  coverage  for  violation  of  the  traffic  regulation
rules</i></s>

                    <s                        pred="http://www.estrellaproject.org/lkif-
core/mereology.owl#contained_in">&lt;v&gt;C1&lt;/v&gt; is contained in &lt;v&gt;A1&lt;/v&gt;</s>

                    <s         pred="&oss;applies">&lt;v&gt;A1&lt;/v&gt;         applies         <i
value="&oss;InsuranceContract">an insurance contract</i></s>

                    &lt;not&gt;&lt;not         assumable="true"&gt;<s         pred="&oss;judged_as"
assumable="true">&lt;v&gt;C1&lt;/v&gt;  is  judged  with  <i  value="&oss;ExtensiveInterpretation">
extensive interpretation</i> </s>

                    &lt;/not&gt;&lt;/not&gt;

                    &lt;not           assumable="true"&gt;<s           pred="&oss;judged_as"
assumable="true">&lt;v&gt;C1&lt;/v&gt;  is not  judged  with  <i  value="&oss;AnalogicInterpretation">
analogic interpretation methods</i> </s>
```

```
</not>

<not><not assumable="true"><s pred="&oss;judged_as" assumable="true"><v>C1</v> is judged as <i value="&oss;Cass.14302/1999"> precedent Cass. 14302/1999</i> </s>

</not></not>

</body>

</rule>

<rule id="JINT_NoCompromissory2">

<head>

<not><s pred="&oss;applies"><v>C1</v> is not <i value="&oss;ArbitrationAgreement">an arbitration clause</i></s></not>

</head>

<body>

<s pred="&oss;hasfactor"><v>C1</v> has factor<i value="&oss;NonRitualArbitration">an irritual arbitration</i></s>

<not><not assumable="true"><s pred="&oss;judged_as" assumable="true"><v>C1</v> is judged as <i value="&oss;Cass.8788/2000"> precedent Cass. 8788/2000</i> </s>

</not></not>

</body>

</rule>

<rule id="JINT_Compromissory">

<head>

<s pred="&oss;applies"><v>C1</v> is <i value="&oss;ArbitrationAgreement">an arbitration clause</i></s>

</head>

<body>

<or><s pred="&oss;hasfactor"><v>C1</v> has factor<i value="&oss;NonRitualArbitration">an irritual arbitration</i></s>

<s pred="&oss;hasfactor"><v>C1</v> has factor<i value="&oss;RitualArbitration">a ritual arbitration</i></s></or>

<s pred="&oss;hasfactor"><v>C1</v> has factor<i value="&oss;PreventsJurisdiction">prevents normal jurisdiction</i></s>

<not><not assumable="true"><s pred="&oss;judged_as" assumable="true"><v>C1</v> is judged as <i value="&oss;Cass.19865/2003"> precedent Cass. 19865/2003</i> </s>

</not></not>

</body>

</rule>
```

```
<rule id="JINT_BindingSurvey">

    <head>

        <s pred="&oss;hasfactor"><v>C1</v> is <i value="&oss;BindingSurvey">an arbitration clause</i></s>

        <not><s pred="&oss;hasfactor"><v>C1</v> is <i value="&oss;NonRitualArbitration">a ritual arbitration clause</i></s></not>

    </head>

    <body>

        <s pred="&oss;hasfactor"><v>C1</v> has factor<i value="&oss;TechnicalEvaluation">involves technical evaluations</i></s>

        <s pred="http://www.estrellaproject.org/lkif-core/mereology.owl#contained_in"><v>C1</v> is contained in <v>A1</v></s>

        <s pred="&oss;applies"><v>A1</v> applies <i value="&oss;InsuranceContract">an insurance contract</i></s>

        <not assumable="true"><s pred="&oss;hasfactor" assumable="true"><v>C1</v> has factor <i value="&oss;LegalEvaluation"> involves legal evaluations</i> </s>

        </not>

        <not><not assumable="true"><s pred="&oss;judged_as" assumable="true"><v>C1</v> is judged as <i value="&oss;Cass.14909/2002"> precedent Cass.14909/2002</i> </s>

        </not></not>

    </body>

</rule>

<rule id="JINT_IrritualArbitration">

    <head>

        <s pred="&oss;hasfactor"><v>C1</v> is <i value="&oss;NonRitualArbitration">an irritual arbitration clause</i></s>

    </head>

    <body>

        <or><s pred="&oss;hasfactor"><v>C1</v> has factor<i value="&oss;LegalEvaluation">involves legal evaluations</i></s>

            <s pred="&oss;hasfactor"><v>C1</v> has factor<i value="&oss;DisputeResolution">involves a dispute resolution</i></s></or>

        <s pred="http://www.estrellaproject.org/lkif-core/mereology.owl#contained_in"><v>C1</v> is contained in <v>A1</v></s>

        <s pred="&oss;applies"><v>A1</v> applies <i value="&oss;InsuranceContract">an insurance contract</i></s>

        <not><not assumable="true"><s pred="&oss;judged_as" assumable="true"><v>C1</v> is judged as <i value="&oss;Cass.9996/2004"> precedent Cass.9996/2004</i> </s>
```

```
                    </not></not>

                </body>

            </rule>

            <rule id="JINT_StatutoryArbitration">

                <head>

                    <s pred="&oss;applies"><v>C1</v> is <i value="&oss;SpecificallySigned">specifically signed</i></s>

                </head>

                <body>

                    <s pred="&oss;applies"><v>C1</v> is <i value="&oss;ArbitrationAgreement">an arbitration clause</i></s>

                    <s pred="http://www.estrellaproject.org/lkif-core/mereology.owl#contained_in"><v>C1</v> is contained in <v>A1</v></s>

                    <s pred="&oss;hasfactor"><v>A1</v> applies <i value="&oss;Statute">a statute</i></s>

                    <not><not assumable="true"><s pred="&oss;judged_as" assumable="true"><v>C1</v> is judged as <i value="&oss;Cass.4351/1993"> precedent Cass. 4351/1993</i> </s>

                    </not></not>

                </body>

            </rule>

        </rules>

    </theory>

<theory id="GeneralClauseRules">

    <imports>

    </imports>

<rules>

<rule id="JINT_KnownDocumentRecall">

  <head>

<not><s pred="&oss;applies"><v>C1</v> is not <i value="&oss;General">a general clause</i></s></not>

</head>

<body>

<s pred="&oss;recalls"><v>C1</v> recalls <v>D1</v> </s>

<s pred="&oss;hasfactor"><v>D1</v> has factor <i value="&oss;KnownToParties"> is known to the parties</i> </s>
```

```
<and assumable="true"><s pred="&oss;judged_as"><v>C1</v> is judged as <i value="&oss;Cass.3929/1999"> precedent Cass. 3929/1999</i> </s></and>

</body>

</rule>

</rules>

</theory>

<theory id="OppressiveRules">

        <imports>

        </imports>

        <rules>

<rule id="JINT_InsuranceCoverageLimitation">

    <head>

        <s pred="&oss;applies"><v>C1</v> is <i value="&oss;LiabilityLimitation">a clause limiting liability</i></s>

    </head>

    <body>

        <s pred="&oss;hasfactor"><v>C1</v> has factor<i value="&oss;ExcludesCoverageForTrafficRegulationViolation">excludes coverage for violation of the traffic regulation rules</i></s>

        <s pred="http://www.estrellaproject.org/lkif-core/mereology.owl#contained_in"><v>C1</v> is contained in <v>A1</v></s>

        <s pred="&oss;applies"><v>A1</v> has factor<i value="&oss;InsuranceContract">an insurance contract</i></s>

        <not><not assumable="true"><s pred="&oss;judged_as" assumable="true"><v>C1</v> is judged as <i value="&oss;TribVeneziaI_Int1"> precedent Trib. Venezia 11 july 2002</i> </s>

        </not></not>

    </body>

</rule>

            <rule id="JINT_RiskExclusionElectricPhenomena">

                <head>

                    <s pred="&oss;hasfactor"><v>C1</v> has factor<i value="&oss;ReducesRiskAssumed"> clause reducing risk taken elsewhere</i></s>

                </head>

                <body>

                    <s pred="&oss;hasfactor"><v>C1</v> has factor<i value="&oss;ExcludesIndirectElectricFenomena">shrinks the risk assumed by the insurance company</i></s>
```

```
                    <s pred="http://www.estrellaproject.org/lkif-core/mereology.owl#contained_in"><v>C1</v> is contained in <v>A1</v></s>

                    <s pred="&oss;contains"><v>A1</v> contains <v>C2</v></s>

                    <s pred="&oss;hasfactor"><v>C2</v> has factor<i value="&oss;CoversLightningDamage">covers damage caused by lightnings</i></s>

                    <not><not assumable="true"><s pred="&oss;judged_as" assumable="true"><v>C1</v> is judged as <i value="&oss;TribTorinoI_Int1"> precedent Trib. Torino gg mm aaaa</i> </s>

                    </not></not>

                </body>

            </rule>

            <rule id="JINT_RiskExclusionElectricDevices">

                <head>

                    <s pred="&oss;hasfactor"><v>C1</v> has factor<i value="&oss;ReducesRiskAssumed"> clause reducing risk taken elsewhere</i></s>

                </head>

                <body>

                    <s pred="&oss;hasfactor"><v>C1</v> has factor<i value="&oss;ExcludesElectronicDevices">shrinks the risk assumed by the insurance company</i></s>

                    <s pred="http://www.estrellaproject.org/lkif-core/mereology.owl#contained_in"><v>C1</v> is contained in <v>A1</v></s>

                    <s pred="&oss;contains"><v>A1</v> contains <v>C2</v></s>

                    <s pred="&oss;hasfactor"><v>C2</v> has factor<i value="&oss;CoversElectricDevices">covers damage caused to electric devices</i></s>

                    <not><not assumable="true"><s pred="&oss;judged_as" assumable="true"><v>C1</v> is judged as <i value="&oss;TribTorinoI_Int1.2"> precedent Trib. Torino  gg mm aaaa</i> </s>

                    </not></not>

                </body>

            </rule>

<rule id="JINT_InsuranceRiskExclusion">

    <head>

        <s pred="&oss;applies"><v>C1</v> is <i value="&oss;LiabilityLimitation">a clause limiting liability</i></s>

    </head>

    <body>

        <s pred="&oss;hasfactor"><v>C1</v> has factor<i value="&oss;ReducesRiskAssumed"> clause reducing risk taken elsewhere</i></s>
```

```
        <s pred="http://www.estrellaproject.org/lkif-
core/mereology.owl#contained_in"><v>C1</v> is contained in <v>A1</v></s>

        <s pred="&oss;applies"><v>A1</v> has factor<i value="&oss;InsuranceContract">an
insurance contract</i></s>

        <not><not assumable="true"><s pred="&oss;judged_as" assumable="true"><v>C1</v>
is judged as <i value="&oss;Cass.816/1979"> precedent Trib. Cass. 816/1979</i> </s>

        </not></not>

    </body>

</rule>

            <rule id="JINT_InsuranceObjectSpecification">

                <head>

                    <s pred="&oss;applies"><v>C1</v> is <i
value="&oss;No_ResponsibilityLimitation">a clause limiting liability</i></s>

                </head>

                <body>

                    <s pred="&oss;hasfactor"><v>C1</v> has factor<i
value="&oss;SpecifyObject"> clause reducing risk taken elsewhere</i></s>

                    <s pred="http://www.estrellaproject.org/lkif-
core/mereology.owl#contained_in"><v>C1</v> is contained in <v>A1</v></s>

                    <s pred="&oss;applies"><v>A1</v> has factor<i
value="&oss;InsuranceContract">an insurance contract</i></s>

                    <not><not assumable="true"><s pred="&oss;judged_as"
assumable="true"><v>C1</v> is judged as <i value="&oss;Cass.16719/2003"> precedent Cass.
16719/2003</i> </s>

                    </not></not>

                </body>

            </rule>

            <rule id="JINT_InsuranceDrunkDriverExclusion">

                <head>

                    <s pred="&oss;hasfactor"><v>C1</v> is <i
value="&oss;SpecifyObject">a clause specifying the object of contract</i></s>

                </head>

                <body>

                    <s pred="&oss;considers"><v>C1</v> has factor<i
value="&oss;DrunkDriver"> drunk driver </i></s>

                    <s pred="http://www.estrellaproject.org/lkif-
core/mereology.owl#contained_in"><v>C1</v> is contained in <v>A1</v></s>

                    <s pred="&oss;applies"><v>A1</v> has factor<i
value="&oss;InsuranceContract">an insurance contract</i></s>
```

```
<not><not assumable="true"><s pred="&oss;judged_as" assumable="true"><v>C1</v> is judged as <i value="&oss;TribTorinoIII_Int1"> precedent Trib. Torino gg mm aaaa</i> </s>

</not></not>

</body>

</rule>



<rule id="JINT_SoleRightInAgency">

<head>

<s pred="&oss;applies"><v>C1</v> is <i value="&oss;ReproducingLawDisposition">a general clause</i></s>

</head>

<body>

<s pred="&oss;hasfactor"><v>C1</v> has factor<i value="&oss;SoleRight">grants exclusive rights</i></s>

<s pred="http://www.estrellaproject.org/lkif-core/mereology.owl#contained_in"><v>C1</v> is contained in <v>A1</v></s>

<s pred="&oss;hasfactor"><v>A1</v> has factor<i value="&oss;AgencyAgreement">agency agreement</i></s>

<not><not assumable="true"><s pred="&oss;judged_as"><v>C1</v> is judged as <i value="&oss;Cass.6369/2001"> precedent Cass. 6369/2001 </i> </s>

</not></not>

<not><not assumable="true"><s pred="&oss;judged_as"><v>C1</v> is judged as <i value="&oss;Cass.14667/2004"> precedent Cass. 14667/2004 </i> </s>

</not></not>

<not><not assumable="true"><s pred="&oss;judged_as"><v>C1</v> is judged as <i value="&oss;Art1743cc"> article 1743 cc</i> </s>

</not></not>

</body>

</rule>

<rule id="JINT_Withdrawal4BothParties">

<head>

<not><s pred="&oss;Oppressive_Status"> <i value="&oss;ContractWithdrawalForBothParties">contract withdrawal for both parties </i></s></not>

</head>

<body>
```

```
                    <s pred="&oss;applies"><v>C1</v> has factor<i value="&oss;ContractWithdrawalForBothParties"> clause reducing risk taken elsewhere</i></s>

                    <s pred="&oss;applies"><v>C1</v> has factor<i value="&oss;Precompiled"> precompiled contract</i></s>

                    <s pred="&oss;applies"><v>C1</v> has factor<i value="&oss;Unilateral"> unilateral contract</i></s>

                    <not><not assumable="true"><s pred="&oss;judged_as" assumable="true"><v>C1</v> is judged as <i value="&oss;Cass.6314/2006"> precedent Cass. 6314/2006</i> </s>

                    </not></not>

                </body>

            </rule>

            <rule id="JINT_ExpertOpinion">

                <head>

                    <not><s pred="&oss;Oppressive_Clause"> <v>C1</v></s></not>

                    <not><s pred="&oss;applies"> <v>C1</v><i value="&oss;CompetenceDerogation"> a comp derogation</i></s></not>

                    <not><s pred="&oss;Oppressive_Status"><i value="&oss;ExpertOpinion"> e</i></s></not>

                </head>

                <body>

                    <s pred="&oss;applies"><v>C1</v> has factor<i value="&oss;ExpertOpinion"> e</i></s>

                    <not><not assumable="true"><s pred="&oss;judged_as" assumable="true"><v>C1</v> is judged as <i value="&oss;Cass.2277/2006"> precedent Cass. 2277/2006</i> </s>

                    </not></not>

                </body>

            </rule>

</rules>

</theory>

<theory id="SpecificSigning">



    <imports>

          </imports>

   <rules>

<rule id="TEC_RecallObjectORNumber">
```

```
  <head>

<s pred="Recalls_ObjectORNumber"><v>B1</v> recalls object or number</s>

</head>

<body>

<or>

<s  pred="&oss;hasfactor"><v>B1</v>  has  factor  <i  value="&oss;RecallsNumber">  recalls
number</i> </s>

<s  pred="&oss;hasfactor"><v>B1</v>  has  factor  <i  value="&oss;RecallsObject">  recalls
object</i> </s>

</or>

</body>

</rule>

<rule id="TEC_RecallObjectANDNumber">

  <head>

<s pred="Recalls_ObjectANDNumber"><v>B1</v> recalls object and number</s>

</head>

<body>

<s  pred="&oss;hasfactor"><v>B1</v>  has  factor  <i  value="&oss;RecallsNumber">  recalls
number</i> </s>

<s  pred="&oss;hasfactor"><v>B1</v>  has  factor  <i  value="&oss;RecallsObject">  recalls
object</i> </s>

</body>

</rule>


<rule id="TEC_CorrectObjectORNumber">

 <head>

         <s pred="CorrectlyRecalled"><v>C1</v> is correctly recalled</s>

</head>

<body>

<s pred="&oss;recalled_by"><v>C1</v> recalled by <v>B1</v> </s>

<s pred="Recalls_ObjectORNumber"><v>B1</v> recalls object or number</s>

<not exception="true"><s pred="required">it is required <i value="ObjectANDNumber"> for
both object and number to be recalled</i></s></not>

</body>

      </rule>
```

```
<rule id="TEC_CorrectObjectANDNumber">

 <head>

         <s pred="CorrectlyRecalled"><v>C1</v> is correctly recalled</s>

</head>

<body>

<s pred="&oss;recalled_by"><v>C1</v> recalled by <v>B1</v> </s>

<s pred="Recalls_ObjectANDNumber"><v>B1</v> recalls object and number</s>

</body>

      </rule>


<rule id="JINT_ObjectANDNumberRequired">

 <head>

         <s pred="required">it is required <i value="ObjectANDNumber"> for both object
and number to be recalled</i></s>

</head>

<body>

<s pred="&oss;applies"><v>C1</v>applies <i value="&oss;Cass.6976/1995">precedent Cass.
6976/1995</i></s>

</body>

      </rule>

<rule id="JINT_ObjectANDNumberNOTRequired">

 <head>

<not><s pred="required">it is required <i value="ObjectANDNumber"> for both object and
number to be recalled</i></s></not>

</head>

<body>

<s pred="&oss;applies"><v>C1</v>applies <i value="&oss;Cass.1317/1998">precedent Cass.
1317/1998</i></s>

</body>

      </rule>

<rule id="JINT_RecallAllClauses">

 <head>

<not><s pred="&oss;applies"><v>C1</v> doesn't apply <i value="&oss;SpecificallySigned">
specifically signed status </i></s></not>

<s pred="RecallException"> <v>C1</v> is subject to the exception</s>
```

```
</head>

<body>

<s pred="&oss;recalled_by"><v>C1</v> recalled by <v>B1</v> </s>

<s pred="&oss;hasfactor"><v>B1</v> has factor <i value="&oss;RecallsAllClauses"> recalls all clauses</i></s>

<not><not assumable="true"><s pred="&oss;applies"><v>C1</v> applies <i value="&oss;Cass.24262/2008">precedent Cass. 24262/2008</i></s></not>

</not></body>

</rule>

<rule id="JINT_RecallNonOppressiveClauses">

 <head>

<not><s pred="&oss;applies"><v>C1</v> doesn't apply <i value="&oss;SpecificallySigned"> specifically signed status </i></s></not>

<s pred="RecallException"> <v>C1</v> is subject to the exception</s>

</head>

<body>

<s pred="&oss;recalled_by"><v>C1</v> recalled by <v>B1</v> </s>

<s pred="&oss;hasfactor"><v>B1</v> has factor <i value="&oss;RecallsNonOppressiveClauses"> recalls also non oppressive clauses</i></s>

<s pred="&oss;judged_as"><v>C1</v> applies<i value="&oss;Cass.5860/1998">precedent Cass.5860/1998</i></s>

</body>

      </rule>

<rule id="TEC_SpecificSigningByRecall">

<head>

<s pred="&oss;applies"><v>C1</v> applies <i value="&oss;SpecificallySigned"> specifically signed status </i></s>

</head>

<body>

<s pred="CorrectlyRecalled"><v>C1</v> is correctly recalled</s>

<not exception="true"><s pred="RecallException"> <v>C1</v> is subject to the exception</s></not>

</body>

</rule>

<rule id="JINT_SignAtEndOfPage">

 <head>
```

```
<not><s pred="&oss;applies"><v>C1</v> doesn't apply <i value="&oss;SpecificallySigned"> specifically signed status </i></s></not>

</head>

<body>

<s pred="&oss;hasfactor"><v>C1</v> has factor <i value="&oss;SigningAtEndOfPage"> is signed at the end of page</i></s>

<not><not assumable="true"><s pred="valid">The <i value="&oss;Cass.4793/2000">precedent Cass. 4793/2000</i> is accepted</s></not>

</not><not exception="true"><s pred="CorrectlyRecalled"><v>C1</v> is correctly recalled</s></not>

</body>

      </rule>

<rule id="JINT_ProducedDuringTrial">

 <head>

<s pred="&oss;applies"><v>C1</v> applies <i value="&oss;SpecificallySigned"> specifically signed status </i></s>

</head>

<body>

<s pred="&oss;hasfactor"><v>C1</v> has factor <i value="&oss;ProducedDuringTrial"> is produced during trial</i></s>

<not><not assumable="true"><s pred="&oss;judged_as"><v>C1</v> is judged as with <i value="&oss;Cass.460/1983">precedent Cass. 460/1983</i> is accepted</s> </not>

</not><not exception="true"><s pred="&oss;hasfactor"><v>C1</v> has factor<i value="&oss;ProducedNotToEnact"> was produced with the intention non to enact it</i></s></not>

</body>

</rule>


 <rule id="JINT_KnownDocument">

  <head>

   <s pred="&oss;hasfactor"><v>C1</v> has factor <i value="&oss;KnownToParties"> is known to the parties</i> </s>

   <s pred="&oss;applies"><v>C1</v> has factor <i value="&oss;Knowledgeable"> is known to the parties</i> </s>

  </head>

  <body>

       <and><s pred="&oss;recalled_by"><v>C1</v> is recalled by <v>D1</v></s>

        <s pred="&oss;Contractual_Agreement"><v>D1</v> is a contractual agreement</s></and>
```

```
      <not><not assumable="true"><s pred="&oss;judged_as"><v>C1</v> is judged as <i value="&oss;TribPiacenzaI_Int2.1"> precedent Trib. Piacenza gg mm aaaa</i> </s></not>

   </not></body>

 </rule>

<rule id="JINT_KnownDocumentRecall">

  <head>

<s pred="&oss;applies"><v>C1</v> is not <i value="&oss;Specific">a general clause</i></s>

</head>

<body>

 <or> <s pred="&oss;hasfactor"><v>C1</v> has factor<i value="&oss;RecallsSeparateDocument">an insurance contract</i></s>

  <and><s pred="&oss;recalls"><v>C1</v> recalls <v>D1</v></s><s pred="&oss;SeparateDocument"><v>D1</v> is a separate doc</s></and>

 </or>

 <s pred="&oss;hasfactor"><v>D1</v> has factor <i value="&oss;KnownToParties"> is known to the parties</i> </s>

<not><not assumable="true"><s pred="&oss;judged_as"><v>C1</v> is judged as <i value="&oss;Cass.3929/1999"> precedent Cass. 3929/1999</i> </s></not>

</not></body>

</rule>

<rule id="JINT_InsuranceDocumentRecall">

  <head>

<not><s pred="&oss;applies"><v>C1</v> is not <i value="&oss;Knowledgeable">knowledgeable</i></s></not>

</head>

<body>

 <or> <s pred="&oss;hasfactor"><v>C1</v> has factor<i value="&oss;RecallsSeparateDocument">an insurance contract</i></s>

  <and><s pred="&oss;recalls"><v>C1</v> recalls <v>D1</v></s><s pred="&oss;SeparateDocument"><v>D1</v> is a separate doc</s></and>

  </or>

 <s pred="http://www.estrellaproject.org/lkif-core/mereology.owl#contained_in"><v>C1</v> is contained in <v>A1</v></s>

<s pred="&oss;hasfactor"><v>A1</v> has factor<i value="&oss;InsuranceContract">an insurance contract</i></s>

<not><not assumable="true"><s pred="&oss;judged_as" assumable="true"><v>C1</v> is judged as <i value="&oss;TribVeneziaI_Int1"> precedent Trib. Venezia 11 july 2002</i> </s>

</not></not>
```

```
</body>

</rule>

</rules>

  </theory>

<theory id="UnilateralPrecompiledRules">

        <imports>

        </imports>

        <rules>

<rule id="TEC_Precompiled">

    <head>

        <s pred="&oss;applies"><v>A1</v> is <i value="&oss;Precompiled">a precompiled
contract</i></s>

    </head>

    <body>

        <s pred="&oss;hasfactor"><v>A1</v> was build for<i
value="&oss;MoreAcceptants">more acceptants</i></s>

        <s pred="&oss;hasfactor"><v>A1</v> was build for<i value="&oss;OneProponent">one
proponent</i></s>

    </body>

</rule>

<rule id="TEC_NoPrecompiled-Contract">

    <head>

        <not><s pred="&oss;applies"><v>A1</v> is <i value="&oss;Precompiled">a not
precompiled contract</i></s></not>

    </head>

    <body>

        <s pred="&oss;hasfactor"><v>A1</v> was build for<i value="&oss;OneAcceptant">one
acceptant</i></s>

        <s pred="&oss;hasfactor"><v>A1</v> was build for<i
value="&oss;MoreProponents">more proponents</i></s>

    </body>

</rule>

<rule id="TEC_NoUnilateral-Clause">

    <head>

        <not><s pred="&oss;applies"><v>C1</v> is <i value="&oss;Unilateral">a unilateral
clause</i></s></not>
```

```
    </head>

    <body>

        <s pred="http://www.estrellaproject.org/lkif-core/mereology.owl#contained_in"><v>C1</v> is contained in <v>A1</v></s>

        <or><and><s pred="&oss;hasfactor"><v>C1</v> was build for<i value="&oss;OneAcceptant">one acceptant</i></s>

            <s pred="&oss;hasfactor"><v>C1</v> was build for<i value="&oss;MoreProponents">more proponents</i></s></and>

            <s pred="&oss;applies"><v>A1</v> is <i value="&oss;NotPrecompiled">a not precompiled contract</i></s></or>

    </body>

</rule>

<rule id="TEC_NoUnilateral-Prepared">

    <head>

        <not><s pred="&oss;applies"><v>C1</v> is <i value="&oss;Unilateral">a unilateral clause</i></s></not>

    </head>

    <body>

        <s pred="http://www.estrellaproject.org/lkif-core/expression.owl#held_by"><v>C1</v> is held by in <v>P1</v></s>

        <s pred="&oss;hasfactor"><v>P1</v> is <i value="&oss;Proposer">the proposer</i></s>

        <s pred="&oss;hasfactor"><v>P1</v> is <i value="&oss;Customer">the customer</i></s>

        <s pred="&oss;hasfactor"><v>P1</v> is <i value="&oss;Occasional">occasional</i></s>

        <s pred="http://www.estrellaproject.org/lkif-core/expression.owl#held_by"><v>C1</v> is held by in <v>P2</v></s>

        <s pred="&oss;hasfactor"><v>P2</v> is <i value="&oss;Contractor">the contractor</i></s>

        <s pred="&oss;hasfactor"><v>P2</v> is <i value="&oss;Professionist">a professionist</i></s>

    </body>

</rule>

            <rule id="JINT_GeneralClauseSubsumption">

                <head>

                    <s pred="&oss;applies"><v>C1</v> is <i value="&oss;General">a general clause</i></s>

                </head>
```

```
<body>

<s pred="&oss;applies"><v>C1</v> has factor<i value="&oss;Unilateral">is unilateral</i></s>

<s pred="http://www.estrellaproject.org/lkif-core/mereology.owl#contained_in"><v>C1</v> is contained in <v>A1</v></s>

<s pred="&oss;applies"><v>A1</v> has factor<i value="&oss;Precompiled">precompiled</i></s>

<not exception="true"><s pred="&oss;applies"><v>C1</v> has factor<i value="&oss;AddedToPrecompiled">is added to precompiled</i></s></not>

<not><not assumable="true"><s pred="&oss;judged_as" assumable="true"><v>C1</v> is judged as <i value="&oss;Cass.13605/1999"> precedent Cass. 13605/1999</i> </s>

</not></not>

</body>

</rule>

<rule id="JINT_PerAdesione-consequence">

<head>

<s pred="&oss;applies"><v>C1</v> is <i value="&oss;Unilateral">a unilateral clause</i></s>

</head>

<body>

<s pred="http://www.estrellaproject.org/lkif-core/mereology.owl#contained_in"><v>C1</v> is contained in <v>A1</v></s>

<s pred="&oss;hasfactor"><v>A1</v> is <i value="&oss;Per_Adesione"></i></s>

<not><not assumable="true"><s pred="&oss;judged_as" assumable="true"><v>A1</v> is judged as <i value="&oss;TribTorinoVIII_Int1"> precedent </i> </s>

</not></not>

</body>

</rule>

<rule id="JINT_NotPerAdesione-consequence">

<head>

<not><s pred="&oss;applies"><v>C1</v> is <i value="&oss;Unilateral">a unilateral clause</i></s></not>

</head>

<body>

<s pred="http://www.estrellaproject.org/lkif-core/mereology.owl#contained_in"><v>C1</v> is contained in <v>A1</v></s>
```

```
<or><not><s pred="&oss;applies"><v>A1</v> is <i value="&oss;Per_Adesione">agreed through mail</i></s></not>

<s pred="&oss;applies"><v>A1</v> is <i value="&oss;NotPer_Adesione">agreed through mail</i></s>

<not><s pred="&oss;hasfactor"><v>A1</v> is <i value="&oss;Per_Adesione">agreed through mail</i></s></not>

<s pred="&oss;hasfactor"><v>A1</v> is <i value="&oss;NotPer_Adesione">agreed through mail</i></s></or>

<not><not assumable="true"><s pred="&oss;judged_as" assumable="true"><v>A1</v> is judged as <i value="&oss;Cass.12153/2006"> precedent Cass. 12153/2006</i> </s>

</not></not>

</body>

</rule>

<rule id="JINT_PerAdesione-causes">

<head>

<s pred="&oss;hasfactor"><v>A1</v> is <i value="&oss;Per_Adesione">a unilateral clause</i></s>

</head>

<body>

<not exception="true"> <s pred="&oss;hasfactor"><v>A1</v> is <i value="&oss;AgreedThroughNegotiation">agreed through direct negotiations</i></s></not>

<s pred="&oss;hasfactor"><v>A1</v> is <i value="&oss;GenericTerms">agreed through mail</i></s>

<s pred="&oss;hasfactor"><v>A1</v> is <i value="&oss;ObjectSoldToThird">agreed through mail</i></s>

<not><not assumable="true"><s pred="&oss;judged_as" assumable="true"><v>A1</v> is judged as <i value="&oss;TribTorinoVIII_Int1"> precedent Trib Torino gg mm aaaa</i> </s>

</not></not>

</body>

</rule>

<rule id="JINT_NotAdesione_Negotiations">

<head>

<not><s pred="&oss;hasfactor"><v>A1</v> is <i value="&oss;Per_Adesione">a Per Adesione contract</i></s></not>

</head>

<body>

<s pred="&oss;hasfactor"><v>A1</v> is <i value="&oss;AgreedThroughNegotiation">agreed through direct negotiations</i></s>
```

```
<not><not assumable="true"><s pred="&oss;judged_as" assumable="true"><v>A1</v> is judged as <i value="&oss;Cass.11757/2006"> precedent Cass 11757/2006</i> </s>

</not></not>

</body>

</rule>

<rule id="JINT_NotPerAdesione-cause">

<head>

<not><s pred="&oss;hasfactor"><v>A1</v> is <i value="&oss;Per_Adesione">a unilateral clause</i></s></not>

</head>

<body>

<s pred="&oss;hasfactor"><v>A1</v> is <i value="&oss;HasExtrahordinaryObject"></i></s>

<s pred="&oss;hasfactor"><v>A1</v> is <i value="&oss;HasRelevantValue"></i></s>

<not><not assumable="true"><s pred="&oss;judged_as" assumable="true"><v>A1</v> is judged as <i value="&oss;CdAMilanoII_Int1"> precedent CdA Milano gg mm aaaa</i> </s>

</not></not>

</body>

</rule>

<rule id="JINT_LAWEXCL_Art1341Exclusion">

<head>

<not><s pred="Relevant_ExArt1341co2"><v>C1</v> falls under the discipline of Article 1341 comma 2 of Civil Code </s></not>

</head>

<body>

<s pred="http://www.estrellaproject.org/lkif-core/mereology.owl#contained_in"><v>C1</v> is contained in <v>A1</v></s>

<not> <s pred="&oss;hasfactor"><v>A1</v> is <i value="&oss;Per_Adesione">agreed through direct negotiations</i></s></not>

<not><not assumable="true"><s pred="&oss;judged_as" assumable="true"><v>A1</v> is judged as <i value="&oss;Cass.12153/06"> precedent Cass. 12153/2006</i> </s>

</not></not>

</body>

</rule>

</rules>
```

```
</theory>

</lkif>
```

a.2.2. Rules in LegalRuleML language

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE Document [

        <!ENTITY xs "http://www.w3.org/2001/XMLSchema">

        <!ENTITY ruleml "http://ruleml.org/">

        <!ENTITY dfs "http://ruleml.org/dfs/">

        <!ENTITY legalruleml "http://legalruleml.org/legalruleml/">

        <!ENTITY dfsont "/ontology/defeasible.owl#">

        <!ENTITY lkif "http://legalruleml.example.org/legalOntology/lkif-core/">

        <!ENTITY                                                          lkif-clo
"http://www.semanticweb.org/ontologies/Judging_Contracts_Core.owl#">

        <!ENTITY                                                          lkif-dlo
"http://www.semanticweb.org/ontologies/Judging_Contracts_Domain.owl#">

        <!ENTITY jurisdictions "http://example.org/jurisdictions.owl#">

        <!ENTITY authorities "http://example.org/authorities.owl#">

        <!ENTITY ACTcc "http://akomantoso.org/it/act/cc/1942-03-16/262/ita/main">

        <!ENTITY      cassprecedent1      "http://akomantoso.org/it/precedent/cc/1981-08-
01/3681/main">

        <!ENTITY      cassprecedent2      "http://akomantoso.org/it/precedent/cc/1994-05-
06/4423/main">

        <!ENTITY      cassprecedent3      "http://akomantoso.org/it/precedent/cc/1998-02-
09/1317main">

        <!ENTITY      cassprecedent1      "http://akomantoso.org/it/precedent/cc/1998-06-
12/5860/main">

        <!ENTITY unibo "http://www.cirsfid.unibo.it/ontology/">

]>

<!--<?xml-model               href="http://ruleml.org/1.0/relaxng/naffologeq_relaxed.rnc"
type="application/relax-ng-compact-syntax"?>-->

<?xml-model        href="../../WP1/schema/nrc_mirror/1.0/relaxng/legalruleml_ordered.rnc"
type="application/relax-ng-compact-syntax"?>
```

<RuleML xmlns="http://ruleml.org/spec" xmlns:dfs="&dfs;" xmlns:xsi="&xs;-instance" xmlns:xs="&xs;" xmlns:lrml="&legalruleml;" xmlns:dc="&dc;" xmlns:dcterms="&dcterms;" xmlns:unibo="&unibo;" xml:base="http://legalruleml.example.org/example/">

<!-- The present collection of 5 rules represents two norms coming from the law (rule1 and rule2) and 3 judicial interpretations carried out by the Cassation Court (rule3, rule4, rule5). interpretation 3 has a head which is the opposite of the head of rules 1 and 2.

The conflict is resolved with priority relations ovr1 and ovr2. Also rules 4 and 5 have inconsistent heads, and a different priority relation (ovr3) manages that too.

Art. 1341 C.C.

Condizioni generali di contratto.

[I] (rule1) Le condizioni generali di contratto predisposte da uno dei contraenti sono efficaci nei confronti dell'altro, se al momento della conclusione del contratto questi le ha conosciute o avrebbe dovuto conoscerle usando l'ordinaria diligenza.

[II] (rule2)In ogni caso non hanno effetto, se non sono specificamente approvate per iscritto, le condizioni che stabiliscono, a favore di colui che le ha predisposte, limitazioni di responsabilitï¿½facoltï¿½i recedere dal contratto o di sospenderne l'esecuzione, ovvero sanciscono a carico dell'altro contraente decadenze, limitazioni alla facoltï¿½i opporre eccezioni, restrizioni alla libertï¿½ontrattuale nei rapporti coi terzi, tacita proroga o rinnovazione del contratto, clausole compromissorie o deroghe alla competenza dell'autoritï¿½iudiziaria.

Art. 1341 Italian Civil Code

General contract conditions.

[I](rule1) General contract conditions written by only one of the parties are efficacious towards the other party only if at the time of the conclusion of the contract he had knowledge of them or could have known them using ordinary diligence.

[II](rule2)In any case are inefficacious, if not specifically accepted by writing, conditions establishing, in favor of the proponent: responsibility limitations, the capacity to withdraw from the contract or to suspend execution; or stating against the other party: disqualifications, limitations to exceptions, contractual restrictions towards third parties, prorogation or renewal of the contract, arbitration agreements or competence derogations.

Cass. Civ. Sez. I 8 giugno 1981, n. 3681.

(rule3) La clausola, secondo la quale in difetto di contestazione l'estratto conto si intende approvato, non puðŸ¤³sere considerata vessatoria, con le conseguenze di cui all'art. 1341, comma 2 c.c., poiche' essa non fa che riprodurre il disposto dell'art. 1832 dello stesso codice.

Cassation Court (Civil Division I) sentence n. 3681 of june 8th 1981.

(rule3) The clause by which in case of dispute the account stated is intended as approved cannot be considered oppressive, with the consequence of article 1341 comma 2 civil code, because it does nothing else than reproducing the disposition of article 1832 of Civil Code.

Cass. Civ. Sez. III 6 maggio 1994, n. 4423

(rule3) In tema di assicurazione della responsabilitï¿½ivile derivante dalla circolazione di veicoli a motore, la clausola che subordina la copertura assicurativa alla circostanza che il conducente dell'automezzo assicurato abbia conseguito la patente di guida, poiche' Ã¨ applicazione di una disposizione di legge, non assume natura vessatoria e non necessita della specifica approvazione scritta ex art. 1341, comma 2, c.c..

Cassation Court (Civil Division III) sentence n. 4423 of may 6th 1994.

(rule3) In the subject of puclic-liability insurance for driving motor vehicles, the clause subordinating the insurance coverage to the circumstance that the driver of the insured vehicles has a driving licence, because it applies a law disposition, is not oppressive and does not need specific acceptation by writing.

Cass. Civ. Sez. III 9 febbraio 1998, n. 1317

(rule4) Agli effetti previsti dall'art. 1341 c.c., e ' ufficiente che la sottoscrizione delle clausole onerose sia apposta dopo un'indicazione idonea a suscitare attenzione, quale quella che richiama il numero o il contenuto delle singole clausole, anche se individuate con riferimento al numero d'ordine o lettera ed all'oggetto di ciascuna di essa.

Cassation Court (Civil Division III) sentence n. 1317 of february 9th 1998.

(rule4) For the purposes of article 1341 c. c., it is sufficient that the specifical signing of oppressive clauses is affixed after an indication sufficient to draw attention, such as that recalling the number or the content of single clauses, even if individuated through the reference to the number or the letter or the object of each of them.

Cass. Civ. Sez. II 12 giugno 1998, n. 5860

(rule5)Non sussiste il requisito della specifica approvazione - a pena di nullitï¿½ilevabile d'ufficio - della deroga convenzionale alla competenza territoriale a favore di un foro esclusivo se la sottoscrizione apposta sul modulo prestampato richiama genericamente gli art. 1341, 1342 c.c. per tutte le condizioni generali di contratto, senza distinzione tra vessatorie e non.

Cassation Court (Civil Division II) sentence n. 5860 of june 12th 1998.

(rule5)The requirement of specific signing is not met Â– the nullity being raised automatically Â– for derogation of competence if the signing affixed on the precompiled module recalling article 1341, 1341 c.c. for all general conditions of contracts, without distinguishing between oppressive and non-oppressive clauses.-->


```
<Assert>

       <lrml:metadata xmlns="http://legalruleml.example.org/nome-del-caso/">

              <lrml:identifications>

                     <lrml:identification id="aut1" iri="&unibo;/person.owl#m.ceci" as="author"/>

                     <lrml:identification id="aut2" iri="&unibo;/person.owl#m.palmirani" as="co-author"/>

                     <lrml:identification id="parliament" iri="&unibo;/organization.owl#ItalianParliament.owl" as="legislator"/>

              </lrml:identifications>

              <lrml:references id="referenceBlock1">

                     <lrml:reference id="art1341" iri="civilcode#art1341" refType="&lkif;#Code"/>

                     <lrml:reference id="art1341-com1" iri="&ACTcc;#art1341-co1" refType="&lkif;#Code"/>
```

```
                    <lrml:reference    id="art1341-com2"    iri="&ACTcc;#art1341-co2"
refType="&lkif;#Code"/>

                    <lrml:reference     id="cass3681-1981"     iri="&cassprecedent1;"
refType="&lkif;#Precedent"/>

                    <lrml:reference     id="cass4423-1994"     iri="&cassprecedent2;"
refType="&lkif;#Precedent"/>

                    <lrml:reference     id="cass1317-1998"     iri="&cassprecedent3;"
refType="&lkif;#Precedent"/>

                    <lrml:reference     id="cass5860-1998"     iri="&cassprecedent4;"
refType="&lkif;#Precedent"/>

              </lrml:references>

              <lrml:events>

                     <!-- events realted to the creation of the rule-->

                     <lrml:event id="e1" value="2012-06-25T01:01:00.0Z"/>

                     <!-- event of efficacy of the code art. 1341 -->

                     <lrml:event id="e2" value="1942-04-21T01:01:00.0Z"/>

                     <!-- events realted to the precedent's applicability-->

                     <lrml:event id="e3" value="1981-08-01T01:01:00.0Z"/>

                     <lrml:event id="e4" value="1994-05-06T01:01:00.0Z"/>

                     <lrml:event id="e5" value="1998-02-09T01:01:00.0Z"/>

                     <lrml:event id="e6" value="1998-06-12T01:01:00.0Z"/>

              </lrml:events>

              <lrml:timesInfo>

                     <lrml:times id="t1">

                            <lrml:time start="#e2" refType="&lkif;#efficacy"/>

                     </lrml:times>

                     <lrml:times id="t2">

                            <lrml:time start="#e3" refType="&lkif;#applicability"/>

                     </lrml:times>

                     <lrml:times id="t3">

                            <lrml:time start="#e5" refType="&lkif;#applicability"/>

                     </lrml:times>

                     <lrml:times id="t4">

                            <lrml:time start="#e6" refType="&lkif;#applicability"/>

                     </lrml:times>
```

```
</lrml:timesInfo>
<lrml:ruleInfo id="ruleInfo1" appliesTo="#rule1">
        <lrml:sources id="sourceBlock1">
                <lrml:source element="#rule1" idRef="#art1341-com1"/>
                <lrml:source element="#rule1" idRef="#art1341"/>
        </lrml:sources>
        <lrml:strength iri="&dfsont;defeasible"/>
        <lrml:jurisdiction iri="&jurisdictions;italy"/>
        <lrml:author idRef="#aut1"/>
        <lrml:times idRef="#t1"/>
        <lrml:creationDateTime idRef="#e1"/>
        <lrml:authority idRef="#Parliament"/>
</lrml:ruleInfo>
<lrml:ruleInfo id="ruleInfo2" appliesTo="#rule2">
        <lrml:sources id="sourceBlock2">
                <lrml:source element="#atom1" idRef="#art1341-com2"/>
                <lrml:source element="#atom2" idRef="#art1341-com1"/>
                <lrml:source element="#atom3" idRef="#art1341-com1"/>
                <lrml:source element="#atom4" idRef="#art1341com2"/>
                <lrml:source element="#atom5" idRef="#art1341com2"/>
                <lrml:source element="#atom5" idRef="#art1341com2"/>
        </lrml:sources>
        <lrml:strength iri="&dfsont;defeater"/>
        <lrml:jurisdiction iri="&jurisdictions;italy"/>
        <lrml:author idRef="#aut1"/>
        <lrml:times irRef="#t1"/>
        <lrml:creationDateTime idRef="#e1"/>
</lrml:ruleInfo>
<lrml:ruleInfo id="ruleInfo3" appliesTo="#rule3">
        <lrml:strength iri="&dfsont;defeater"/>
        <lrml:sources id="sourceBlock3">
                <lrml:source element="#rule3" idRef="#cass3681-1981"/>
                <lrml:source element="#rule3" idRef="#cass4423-1994"/>
```

```
                </lrml:sources>

                <lrml:jurisdiction iri="&jurisdictions;italy"/>

                <lrml:author idRef="#aut1"/>

                <lrml:times idRef="#t2" appliesTo="#atom1"/>

                <lrml:authority idRef="#cassationCourt"/>

                <lrml:creationDateTime idRef="e2"/>

            </lrml:ruleInfo>

            <lrml:ruleInfo id="ruleInfo4" appliesTo="#rule4">

                <lrml:strength iri="&dfs;defeater"/>

                <lrml:sources id="sourceBlock4">

                    <lrml:source element="#rule4" idRef="#cass1317-1998"/>

                </lrml:sources>

                <lrml:jurisdiction iri="&jurisdictions;italy"/>

                <lrml:author idRef="#aut1"/>

                <lrml:times idRef="#t3"/>

                <lrml:authority idRef="#cassationCourt"/>

            </lrml:ruleInfo>

            <lrml:ruleInfo id="ruleInfo5" appliesTo="#rule5">

                <lrml:sources id="sourceBlock5">

                    <lrml:source element="#rule4" idRef="#cass5860-1998"/>

                </lrml:sources>

                <lrml:jurisdiction iri="&jurisdictions;italy"/>

                <lrml:author idRef="#aut1"/>

                <lrml:times idRef="#t4"/>

                <lrml:authority idRef="#cassationCourt"/>

            </lrml:ruleInfo>

        </lrml:metadata>

        <Rulebase node="#rulebase1" mapClosure="universal">

            <implies id="rule1">

                <!-- Art. 1341 Italian Civil Code

General contract conditions.

[I]General contract conditions written by only one of the parties are efficacious
towards the other party only if at the time of the conclusion of the contract he had
knowledge of them or could have known them using ordinary diligence.-->
```

```
<then>
    <Atom>
        <Rel iri="&lkif-clo;considered_by">considered by</Rel>
        <Var type="&lkif-dlo;Contract_Clause">Contract_Clause</Var>
        <Ind iri="#art1341-com1"/>
    </Atom>
</then>
<if>
    <And>
    <Atom>
            <Rel iri="&lkif-clo;applies">applies</Rel>
            <Var type="&lkif-dlo;Contract_Clause">Contract_Clause</Var>
            <Ind iri="&lkif-dlo;General"> General</Ind>
        </Atom>
        <Atom>
            <Rel iri="&lkif-clo;applies">applies</Rel>
            <Var type="&lkif-dlo;Contract_Clause">Contract_Clause</Var>
            <Ind  iri="&lkif-dlo;Unilateral">  Unilateral </Ind>
        </Atom>
        <Not>
            <Atom>
                <Rel iri="&lkif-clo;applies">applies</Rel>
                <Var type="&dlo;Contract_Clause">Contract_Clause</Var>
                <Ind iri="&lkif-dlo;Knowledgeable">Knowledgeable</Ind>
            </Atom>
        </Not>
    </And>
</if>
</implies>
```

```
<!-- Art. 1341 Italian Civil Code

[II]In any case are inefficacious, if not specifically accepted by writing, conditions establishing, in favor of the proponent: responsibility limitations, the capacity to withdraw from the contract or to suspend execution; or stating against the other party: disqualifications, limitations to exceptions, contractual restrictions towards third parties, prorogation or renewal of the contract, arbitration agreements or competence derogations. -->

<implies id="rule2">

<then>

<atom id="atom1">

<Rel iri="&lkif-clo;considered_by">considered
by</Rel>

<Var type="&lkif-
dlo;Contract_Clause">Contract_Clause</Var>

<Ind iri="#art1341-com2"/>

</atom>

</then>

<if>

<And>

<atom id="atom2">

<Rel iri="&lkif-clo;applies">applies</Rel>

<Var type="&lkif-
dlo;Contract_Clause">Contract_Clause</Var>

<Ind iri="&lkif-dlo;General"> General</Ind>

</atom>

<atom id="atom3">

<Rel iri="&lkif-clo;applies">applies</Rel>

<Var type="&lkif-
dlo;Contract_Clause">Contract_Clause</Var>

<Ind iri="&lkif-dlo;Unilateral"> Unilateral
</Ind>

</atom>

<atom id="atom4">

<Rel iri="&lkif-clo;applies">applies</Rel>

<Var type="&lkif-
dlo;Contract_Clause">Contract_Clause</Var>

</atom>

<atom id="atom5">
```

```
<Rel iri="&lkif;#member_of">isA</Rel>
<Var type="&lkif-lo;Contract_Clause">Contract_Clause</Var>
<Ind iri="&lkif-lo;Oppressive_Status">Oppressive_Status</Ind>
</atom>
<Not>
<Atom id="atom6">
<Rel iri="&lkif-clo;applies">applies</Rel>
<Var type="&lkif-dlo;Contract_Clause">Contract_Clause</Var>
<Ind iri="&lkif-dlo;SpecificallySigned">SpecificallySigned</Ind>
</Atom>
</Not>
</And>
</if>
</implies>
<implies id="rule3">
<!-- Cassation Court (Civil Div. I) sentence n. 3681 of june 8th 1981.

The clause by which in case of dispute the account stated is intended as approved cannot be considered oppressive, with the consequence of article 1341 comma 2 civil code, because it does nothing else than reproducing the disposition of article 1832 of Civil Code.

Cassation Court (Civil Div. III) sentence n. 4423 of may 6th 1994.

In the subject of puclic-liability insurance for driving motor vehicles, the clause subordinating the insurance coverage to the circumstance that the driver of the insured vehicles has a driving licence, because it applies a law disposition, is not oppressive and does not need specific acceptation by writing. -->
<then>
<And>
<Not>
<atom id="atom5">
```

```
                                        <Rel
iri="&lkif;#member_of">isA</Rel>

                                        <Var                    type="&lkif-
dlo;Contract_Clause">Contract_Clause</Var>

                                        <Ind                     iri="&lkif-
dlo;Oppressive_Status">Oppressive_Status</Ind>

                                 </atom>

                          </Not>

                          <Not>

                                 <Atom>

                                        <Rel                     iri="&lkif-
clo;considered_by">considered by</Rel>

                                        <Var                    type="&lkif-
dlo;Contract_Clause">Contract_Clause</Var>

                                        <Ind iri="#art1341-com2"/>

                                 </Atom>

                          </Not>

                   </And>

            </then>

            <if>

                   <Atom>

                          <Rel iri="&lkif-clo;applies">applies</Rel>

                          <Var                                  type="&lkif-
dlo;Contract_Clause">Contract_Clause</Var>

                          <Ind                                   iri="&lkif-
dlo;ReproducingLawProvision">ReproducingLawProvision</Ind>

                   </Atom>

            </if>

      </implies>

      <implies id="Rule4">

            <!-- Cassation Court (Civil Div. III) sentence n. 1317 of february
9th 1998.


For the purposes of article 1341 c. c., it is sufficient that the specifical signing of
oppressive clauses is affixed after an indication sufficient to draw attention, such as
that recalling the number or the content of single clauses, even if individuated through
the reference to the number or the letter or the object of each of them. -->

            <then>
```

```
<Atom>
    <Rel iri="&lkif-clo;applies">applies</Rel>
    <Var type="&lkif-dlo;Contract_Clause">Contract_Clause</Var>
    <Ind iri="&lkif-dlo;SpecificallySigned">SpecificallySigned</Ind>
</Atom>
</then>
<if>
    <And>
        <Atom>
            <Rel iri="&lkif-clo;recalled_by">recalled_by</Rel>
            <Var type="&lkif-dlo;Contract_Clause">Contract_Clause</Var>
            <Var type="&lkif-dlo;Contract_Box">Contract_Box</Var>
        </Atom>
        <Atom>
            <Rel iri="&lkif-clo;applies">applies</Rel>
            <Var type="&lkif-dlo;Contract_Clause">Contract_Clause</Var>
            <Ind iri="&lkif-dlo;RecallsObjectAndNumber">RecallsObjectOrNumber</Ind>
        </Atom>
    </And>
</if>
</implies>
<implies id="Rule5">
    <!-- Cassation Court (Civil Div. II) sentence n. 5860 of june 12th 1998.

The requirement of specific signing is not met Â– the nullity being raised automatically Â– for derogation of competence if the signing affixed on the precompiled module recalling article 1341, 1341 c.c. for all general conditions of contracts, without distinguishing between oppressive and non-oppressive clauses. -->
    <then>
        <Not>
```

```
                              <Atom>
                                      <Rel iri="&lkif-clo;applies">applies</Rel>
                                      <Var type="&lkif-dlo;Contract_Clause">Contract_Clause</Var>
                                      <Ind iri="&lkif-dlo;SpecificallySigned">SpecificallySigned</Ind>
                              </Atom>
                      </Not>
              </then>
              <if>
                      <And>
                              <Atom>
                                      <Rel iri="&lkif-clo;recalled_by">recalled_by</Rel>
                                      <Var type="&lkif-dlo;Contract_Clause">Contract_Clause</Var>
                                      <Var type="&lkif-dlo;Contract_Box">Contract_Box</Var>
                              </Atom>
                              <Atom>
                                      <Rel iri="&lkif-dlo;applies">applies</Rel>
                                      <Var type="&lkif-dlo;Contract_Box">Contract_Box</Var>
                                      <Ind iri="&lkif-dlo;RecallsNonOppressiveClauses">RecallsNonOppressiveClauses</Ind>
                              </Atom>
                      </And>
              </if>
      </implies>
      <Overrides id="ovr1">
              <Rule keyref="rule_3"/>
              <Rule keyref="rule_2"/>
      </Overrides>
      <Overrides id="ovr2">
              <Rule keyref="rule_5"/>
              <Rule keyref="rule_4"/>
```

```
            </Overrides>
      </Rulebase>
</Assert>
</RuleML>
```

a.2.3. The rules in Clojure language

```
;;; Copyright (c) 2010 Fraunhofer Gesellschaft
;;; Licensed under the EUPL V.1.1

(ns carneades.engine.test-Art1401
  (:use clojure.test
        carneades.engine.shell
        carneades.engine.argument
        carneades.engine.scheme
        carneades.engine.caes))

(def theory1
  (make-theory
    :sections
       [(make-section
              :header (make-metadata :title "Inefficacy" :creator "Marcello")
       :schemes
       [(make-scheme
          :header (make-metadata :title "General-Inefficacy" :creator "Marcello")
          :conclusions ['(Inefficacious ?x)]
          :premises [(pm '(considered_by ?x ?y))
                                     (pm '(applies ?x Clause_Inefficacy))]
                                      ])]
(def theory2
  (make-theory
    :sections
    [(make-section
```

```
        :header (make-metadata :title "Art1341" :creator "Marcello")
:schemes
[(make-scheme
   :header (make-metadata :title "Art1341co1" :creator "Marcello")
   :conclusions ['(Relevant_ExArt1341co1 ?x)]
   :premises [(pm '(applies ?x General))
                              (pm '(applies ?x Unilateral))
                              (pm '(applies ?x Not_Knowledgeable))])
        (make-scheme
   :header (make-metadata :title "Art1341co2" :creator "Marcello")
   :conclusions ['(Relevant_ExArt1341co2 ?x)]
   :premises [(pm '(applies ?x ?y))
                             (pm '(Oppressive_Status ?y))
                              (pm '(applies ?x General))
                              (pm '(applies ?x Unilateral))
                              (pm '(not(applies ?x SpecificallySigned)))])

                              ])]
 [(make-section
        :header (make-metadata :title "Art1342" :creator "Marcello")
:schemes
[(make-scheme
   :header (make-metadata :title "Art1342co1-1" :creator "Marcello")
   :conclusions ['(Better_ExArt1342co1 ?x)]
   :premises [(pm '(contained_in ?x ?c))
                              (pm '(applies ?c Precompiled))
                              (pm '(applies ?x AddedToPrecompiled))])
        (make-scheme
   :header (make-metadata :title "Art1342co1-1" :creator "Marcello")
   :conclusions ['(Worse_ExArt1342co1 ?x)]
   :premises [(pm '(contained_in ?x ?c))
                              (pm '(applies ?c Precompiled))
                              (pm '(not(applies ?x AddedToPrecompiled)))])
```

```
        (make-scheme
   :header (make-metadata :title "Art1342co2" :creator "Marcello")
   :conclusions ['(Relevant_ExArt1342co2 ?x)]
   :premises [(pm '(contained_in ?x ?c))
                              (pm '(applies ?c Precompiled))
                              (pm '(applies ?x ?y))
                              (pm '(Oppressive_Status ?y))
                              (pm '(not(applies ?x SpecificallySigned)))])

                              ])]
[(make-section ; a GENERAL rule!
        :header (make-metadata :title "Inefficacy" :creator "Marcello")
:schemes
[(make-scheme
   :header (make-metadata :title "General-Inefficacy" :creator "Marcello")
   :conclusions ['(Inefficacious ?x)]
   :premises [(pm '(considered_by ?x ?y))
                             (pm '(applies ?y Clause_Inefficacy))]
                              )])]
[(make-section
        :header (make-metadata :title "Art1341-Consequences" :creator "Marcello")
:schemes
[(make-scheme
   :header (make-metadata :title "Art1341co1-Consequence" :creator "Marcello")
   :conclusions ['(considered_by ?x Art1341co1cc)]
   :premises [(pm '(Relevant_ExArt1341co1 ?x))]
          :exceptions [(pm '(applies ?x International))
                              (pm '(applies ?x ReproducingLawDisposition))])
         (make-scheme
   :header (make-metadata :title "Art1341co2-Consequence" :creator "Marcello")
   :conclusions ['(considered_by ?x Art1342co1cc)]
   :premises [(pm '(Relevant_ExArt1342co1 ?x))]
          :exceptions [(pm '(applies ?x International))
```

```
                                (pm '(applies ?x ReproducingLawDisposition))])])]
[(make-section
        :header (make-metadata :title "Art1342-Consequences" :creator "Marcello")
:schemes
[
   (make-scheme
   :header (make-metadata :title "Art1342co1-Consequence" :creator "Marcello")
   :conclusions ['(normatively_striclty_worse ?x ?y)]
   :premises [(pm '(contained_in ?x ?c))
                                (pm '(contained_in ?y ?c))
                                (pm '(Better_ExArt1342co1 ?x))
                                (pm '(Worse_ExArt1342co1 ?y))])
   (make-scheme
   :header (make-metadata :title "Art1342co2-Consequence" :creator "Marcello")
   :conclusions ['(considered_by ?x Art1342co2cc)]
   :premises [(pm '(Relevant_ExArt1342co2 ?x))]
          :exceptions [(pm '(applies ?x International))
                                (pm '(applies ?x ReproducingLawDisposition))])

                                ])]
[(make-section
        :header (make-metadata :title "Art1341co1-Exclusion" :creator "Marcello")
:schemes
[(make-scheme
   :header (make-metadata :title "Art1341co1-Exclusion-2" :creator "Marcello")
   :conclusions ['(not (Relevant_ExArt1341co1 ?x))]
   :premises [(pm '(not(applies ?x General)))])
   (make-scheme
   :header (make-metadata :title "Art1341co1-Exclusion-3" :creator "Marcello")
   :conclusions ['(not (Relevant_ExArt1341co1 ?x))]
   :premises [(pm '(not(applies ?x Unilateral)))])
        (make-scheme
   :header (make-metadata :title "Art1341-Exclusion-4" :creator "Marcello")
```

```
          :conclusions ['(not (Relevant_ExArt1341co1 ?x))]
          :premises [(pm '(applies ?x Knowledgeable))])
        [(make-section
               :header (make-metadata :title "Art1341co2-Exclusion" :creator "Marcello")
               :schemes
               [(make-scheme
          :header (make-metadata :title "Art1341co2-Exclusion-1" :creator "Marcello")
          :conclusions ['(not (Relevant_ExArt1341co1 ?x))]
          :premises [(pm '(applies ?x SpecificallySigned))])
               (make-scheme
          :header (make-metadata :title "Art1341co2-Exclusion-2" :creator "Marcello")
          :conclusions ['(not (Relevant_ExArt1341co2 ?x))]
          :premises [(pm '(not (Relevant_ExArt1341co1 ?x)))])
                 ])]
                              ))


(def max-goals 500)
(def generators (list (generate-arguments-from-theory theory1)))


(defn ag [facts query]  ;
  "(seq-of literal) literal -> argument-graph
   construct and evaluate an argument graph"
  (argue (make-engine max-goals facts generators)
         carneades-evaluator
         query))


(deftest test-engine-example1
         (let    [facts    '((not    (applies    AOUMessina/ByteSoftwareHouse_Clause2.020
SpecificallySigned))
                              (applies           AOUMessina/ByteSoftwareHouse_Clause2.020
ArbitrationAgreement)
                              (applies AOUMessina/ByteSoftwareHouse_Clause2.020 General)
                              (applies           AOUMessina/ByteSoftwareHouse_Clause2.020
Unilateral)
```

```
                        (applies Art1341co2 Clause_Inefficacy)
                        (Oppressive_Status ArbitrationAgreement)
                        )
        query '(Inefficacious AOUMessina/ByteSoftwareHouse_Clause2.020)]
      (is (in? (ag facts query) query))))

(deftest test-engine-example2
     (let [facts '((not (contained_in AgenziaImmobiliareD/NgPF_ClauseX
AgenziaImmobiliareD/NgPF_Contract))
                        (applies AgenziaImmobiliareD/NgPF_Contract Precompiled)
                        (applies AgenziaImmobiliareD/NgPF_ClauseX
SpecificallySigned)
                        (applies AgenziaImmobiliareD/NgPF_ClauseX Unilateral)
                        )
        query '(not (Relevant_ExArt1341co2 AgenziaImmobiliareD/NgPF_ClauseX))]
      (is (in? (ag facts query) query))))

(deftest test-engine-example3
     (let [facts '((not (contained_in LLoyd/FF_Clause Lloyd/FF_Contract))
                        (applies Lloyd/FF_Contract Precompiled)
                        (applies LLoyd/FF_Clause AddedToPrecompiled)
                        (applies LLoyd/FF_Clause SpecificallySigned)
                        )
        query '(not (Better_ExArt1342co1 LLoyd/FF_Clause))]
      (is (in? (ag facts query) query))))
```