Browsing case-law: an Application of the Carneades Argumentation System

Marcello Ceci[1],Thomas F. Gordon[2]

[1] CIRSFID, University of Bologna, Italy
[2] Fraunhofer-FOKUS Institut, Berlin, Germany

m.ceci@unibo.it
thomas.gordon@fokus.fraunhofer.de

Abstract. This paper presents an application of the Carneades Argumentation System to case-law. The application relies on a set of ontologies – representing the core and domain concepts of a restricted legal field, the law of contracts – and a collection of precedents taken from Italian courts of different grades. The knowledge base represents the starting point for the construction of rules representing laws and precedents which, in turn, are responsible for the argumentative reasoning. The system reconstructs the legal interpretations performed by the judge, presenting its reasoning path and suggesting possible different or divergent interpretation in the light of relevant code- and case-law.

Keywords: AI&Law, Legal Argumentation, Semantic Web, Legal Ontology.

1 Introduction

Precedent is a main element of legal knowledge worldwide: by settling conflicts and sanctioning illegal behaviours, judicial activity enforces law provisions within the national borders, therefore supporting the validity of laws as well as the sovereignty of the government that issued them. Representing the content of case-law, in terms of legal concepts taken into consideration and interpretations performed by the judge, is a very interesting task for the IT research, and the AI & LAW community has presented very significant outcomes in this topic since the '80, with different approaches: legal case-base reasoning (HYPO, CATO, IBP, CABARET), legal concepts representation through logics [1], rule interchange for applications in the legal domain [8] and more recently also argumentation.

1.1. Carneades

Carneades[1] [10] is a set of open source software tools for mapping and evaluating arguments, under development since 2006. Carneades contains a logical model of argumentation based on Doug Walton's theory of argumentation, and developed in

[1] http://carneades.berlios.de

collaboration with him. In particular, it implements Walton's argumentation schemes [11] 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 arguments graphs to represent ongoing dialogues. It can therefore be used for studying argumentation from a computational perspective, but also to develop 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 [14], but it also takes into account the relational conception of argument[2] [4].

1.2. The application to case-law: objectives and methodology

In the present application, Carneades' potentialities will be exploited to conduct reasoning on case-law, whose knowledge has been previously modeled in an OWL/RDF ontology and in a set of rules (in LKIF-Rule language [2]).

The goal of the present approach is to define a framework for case-law semantics, exploiting Semantic Web technologies to "fill the gap" between document representation and rules modelling [13], ensuring isomorphism between the text fragment (the only binding legal expression) and the rule. Cornerstone of the framework is the ontology, intended in its computer science meaning: a shared vocabulary, a taxonomy and axioms which represent a domain of knowledge by defining objects and concepts together with their properties, relations and semantics. We believe that the features of OWL2 could greatly improve legal concepts modelling and reasoning, once properly combined with rule modelling. Our aim is hence to formalize the legal concepts and the argumentation patterns contained in the judgment in order to check, validate and reuse the elements of judgement as expressed by the text and the argumentation contained in it.

To achieve this, four models are necessary:

- a *document metadata structure*, capturing the main parts of the judgment to create a bridge between text and semantic annotation of legal concepts;
- a *legal core ontology*, describing the legal domain's main elements in terms of general concepts through an LKIF-Core extension;
- a *legal domain ontology*, modelling the legal concepts of a specific legal domain concerned by the case-law, including a set of sample precedents;
- *argumentation modelling and reasoning*, representing the structure and dynamics of argumentation.

Three kinds of knowledge are modelled in the ontology and the rules: legal rules (laws and other authoritative acts), case-law (precedents and their relevant interpretations), and material circumstances (the object of the judgment: i.e., a contract clause, a human behaviour, an event).

---

[2] The main difference between the two conceptions 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.

Aim of this application is to create a reasoning environment allowing a high level of human-machine interaction: the user can start from some basic concept (a legal concept, a fact, an exception, a law prescription) and query the database (which is contained in the OWL ontology) to get some “pilot cases” in return. The user can then ask about the outcome of the pilot case, and about the main interpretations made by the judge in his decision: these are presented in a graph which shows not only the logical process followed by the judge and the laws which he applied but also those who could be, and the precedents which – if accepted - could lead to a different judgement.

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 is ought to represent the dynamics of norms and judicial interpretations.

The present approach focuses on the "argument from ontology" feature of Carneades [6]: 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 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, interpretations performed by the judge. The conclusion of the argumentation represents the final adjudication of the claim, and the Carneades reasoner is expected to accept or reject the claim by semantically applying 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, the tool presents to the user a semantic environment where different laws, legal statuses and precedents are semantically 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.

2 The case-law argumentation ontologies

The "Core and Domain Legal Ontologies" are two OWL/RDF ontologies conceived to model the semantics of judicial interpretations, developed in the contest of a research on judicial knowledge modelling [12].

2.1. The Core Ontology

The core ontology (an extension of the LKIF-Core Ontology) introduces the main concepts of the legal domain, defining the classes which will be later filled with the metadata of judicial decisions. Even though the core ontology should be domain-generic and not modeled upon a specific legal subject, the sample model was conceived only to successfully represent the interaction in the civil law subject, when contracts, laws and judicial decisions come into play. An important modeling choice is the reification of legal statuses and the creation of the object property "applies" (subproperty of LKIF's "qualifies") to link a material circumstance to its status.

2.2. The Domain Ontology

Following the structure outlined in the Core Ontology, the metadata taken from judicial documents are represented in the Domain Ontology. The modeling was carried out manually by an expert in the legal subject (a graduated jurist), which actually represents the only viable choice in the legal domain: automatic information retrieval and machine learning techniques, in fact, do not yet ensure a sufficient level of accuracy. Building a 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, while at the same time keeping an open approach. Here, the property "applies" is used to link the fact of the cases to its legal statuses, in such a way as to give evidence to interpretations made by the judge (represented by the object property "judged_as", subproperty of "applies").

2.3. Features of the Ontology Set

The so-built layered ontology creates an 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 decision itself. Example 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 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 possible weak spots in the decision's groundings; suggest 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.

The layered structure of the ontology set allows an efficacious scaling from legal concepts to factors, up to dimensions and legal principles: all these concepts can be represented in the domain ontology, and the hook of the core concepts to LKIF-Core

should ensure a good semantic alignment between different domain ontologies, as far as different authors are concerned (even if, at the present time, this research has not yet evaluated the alignment capabilities of the present ontology set).

2.4. Changes made to the ontology set for the present application

This application of Carneades involved the following modifications for the Core and Domain ontologies [3]:

- enriching the semantic content of the ontology set by representing finer-grained knowledge contained in the decision's text, in an environment where this expansion of the knowledge base does not entail an overloading of the OWL reasoners, which would compromise computability;
- modeling a rule system representing the dynamic relationships created by judicial interpretation and law application;
- importing knowledge from the ontology set in such a way as to allow successful interaction with the rule set and the Carneades model.

3 Constructing Arguments with Carneades

Carneades is a tool which relies on a solid background theory involving Walton's argumentation schemes with an articulated conception of the burden of proof and of its allocation [9]. However, even though the present application takes advantage from the strong conceptual foundations of this tool, these two features will be set aside. The main reason for this is that (as of Carneades version 1.0.2) these functions are not implemented in the automatic argument construction process, which itself represents the pivot of this experiment. Particularly for the burden of proof this is unfortunate, since the present experiment would gain a lot of depth from an automatic management of burden of proofs and proof standards.

While the new version of Carneades (2.x, currently under development) will use the Clojure language, the latest complete version (1.0.2, the one used for the present application) relies on the LKIF-Rule language [7].

The features of Carneades which are thoroughly used in the present application are the ontology import module, the argxument construction module, and the argument visualizer.

3.1. Ontology Import Module

The ontology import module allows the system to automatically import (stated and inferred) knowledge coming from any OWL/RDF ontology [6]. This knowledge is then used by Carneades during its argumentation process, in order to accept those assertions inside the argument graph which are recorded as true into the (stated or inferred) ontology. What the Carneades ontology import feature cannot do, as of version 1.0.2, is to show the reasoning process, followed by the Hermit reasoner[3] to

[3] HermiT is reasoner for ontologies written using the Web Ontology Language. Given an OWL file, a reasoner can perform tasks such as determining whether or not the ontology is consistent, identifying subsumption relationships between classes, inferring new knowledge.

infer those assertions. The ontology import is instead a black box: in the argument visualizer, assertions accepted on the basis of knowledge coming from the ontology actually appear as accepted under the condition that the whole ontology is valid, without any argumentation being provided nor a distinction between stated and inferred knowledge being made. This means that, at the present time, only the reasoning coming from the application of rules to the knowledge base is showed in the visualizer: nevertheless, having solved some modeling issues on this behalf, this is already enough to show the core part of the interpretation process followed by the judge.

3.2. Argument Construction Module

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 [10]. The target statement can be of two types: a "query-like" statement (i.e. ?x Oppressive_Clause, which means “give me 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 towards the desired goal (pro or con, which means towards the acceptance or the rejection of the target statement).

3.3. Argument Visualizer

The argument visualizer is a powerful tool that allows the user to manage the argumentation process (manually adding assumptions, accepting/rejecting statements, cutting out parts of reasoning, and so on) in order to investigate the concepts of the case in details and relevant case law, not by comparing their textual appearance, but rather by manipulating the concepts expressed in them [5].

4 The Scenario: Consumer Law

In this demonstration of Carneades application, we will examine a piece of Italian Consumer Law[4] by:

- retrieving the case-law concerning a legal concept;
- analyzing the interpretations made by the judge on that case;
- searching possible alternative solutions to the case.

Please notice that, in order to simplify the understanding of the graph, in this presentation the search for arguments will be "progressive": this means that we will ask the reasoner to conduct argumentation on statements only to a limited extent, in order to unveil the argumentation graph step by step. This is done by limiting the number of nodes that the reasoner can follow before returning results. By raising this value, it is possible to obtain more detailed or complex graphs in a single step, either

[4] Actually regulated through legislative decree n. 206 of September 6th, 2005 - even though some cases still fall under artt. 1341 and 1342 of Italian Civil Code, which are still in force.

by deepening the search, to resolve exceptions and indirect argumentation paths in a single step, or by broadening it, to search for more "daring" argumentation paths.

4.1. Modelling of the law

Preliminarily, the two norms involved in the reasoning will be presented:

Article 1341 comma 1 of Italian Civil Code – General contract clauses which have been unilaterally predisposed by one of the contract parties are efficacious only if they were known by the other contract party, or knowable by using ordinary diligence.

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 modelling of this information is based on both the ontology and the rules. In particular, the ontology contains "static" information on the law (such as the enacting authority, the subject, the legal concepts contained in the text, the URI of the legal expression), while the rules classify the material circumstances (in this case, the contract clauses) which share certain legal statuses as being relevant under that law. This is an example of a rule stating the relevancy for comma 2 of Article 1341:

```
<!ENTITY oss "http://www.semanticweb.org/ontologies/2011/8/prova1.owl#">
<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>
```

Please notice that the rule does not contain the list of statuses, but rather refers to a class of “Oppressive statuses” (a naming acknowledged by the legal doctrine), whose modelling is left to the ontology (Fig. 1). This distribution in the representation of the law allows an open organization of legal knowledge, while

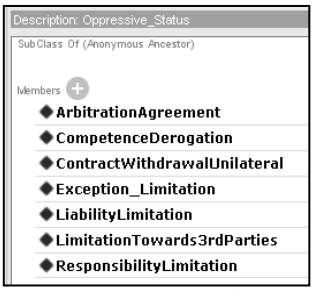


Fig. 1. Members of the Oppressive_Status class

at the same time keeping the full expressivity of the rule syntax.

The rule presented above only states which circumstances are subsumed under that legal rule; successively, another rule comes into play, verifying if any exceptions to the general rule apply. If not, the consequence of the legal rule (in this case, inefficacy) is related to the circumstance (the contract clause):

```
<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>
```

4.2. Modelling of the contract

The material circumstances which are taken into account by a precedent are modelled only in the domain ontology (not in the rules), and semantically classified depending on its characteristics (in the case of a contract clause: containing contract, contract parties, object of the contract, object of the clause). Two different properties represent the relation between a circumstance (the clause) and a legal status: "applies", which means that the status has been recognised by both parties as applicable to the circumstance, and "judged_as", which means that status has been interpreted as being applicable to the circumstance by a judge.

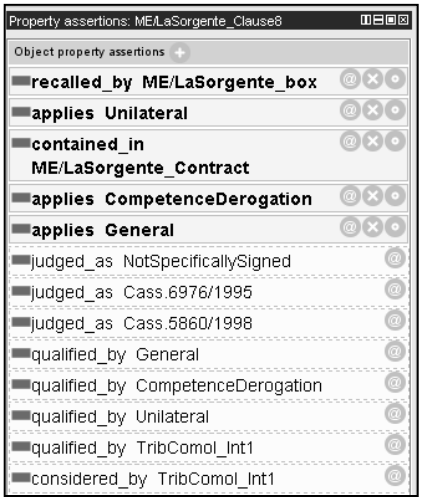


Fig. 2. Properties of a contract clause instance

So, for example, the contract clause ME/LaSorgente_Clause8 (Clause 8 of the contract between M.E. and "La Sorgente") has the characteristics indicated in Fig. 2.

We can see that "Unilateral", "CompetenceDerogation", "General" are three characteristics which are acknowledged by both parties; "recalled_by ME/LaSorgente_box" represents a relation of this clause with another part of the contract; the "considered_by" property links to the precedent (and therefore the authority) which produced the subsumptions: a legal status ("NotSpecificallySigned") and two precedents which have been cited (Cass. 6976/1995 and Cass. 5860/1998). In the next chapter we will see how this knowledge is managed in Carneades. Finally, please notice that the clause also applies the status of "CompetenceDerogation", an Oppressive_Status.

4.3. Modelling of the case-law

The judicial interpretation instances are modelled both into the domain ontology and into the rules. The domain ontology contains static knowledge (enacting authority, object of the case, classification, a URI) as well as "surface" information such as those presented above: the circumstance taken into consideration, the legal status under which the circumstance is subsumed, the precedents cited.

The mechanics underlying the judicial interpretation are contained in the rules. This is an example of a rule representing a judicial subsumption:

```
<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 assumable="true" pred="&oss;judged_as"><v>C1</v> applies<i
        value="&oss;Cass.5860/1998">precedent Cass.5860/1998</i></s>
    </body>
</rule>
```

Please notice the particular role given to the precedent "Cass.5860/1998": it is an assumption, so it does not prevent the system from suggesting this particular interpretation (and the precedent) as a result of the reasoning process on cases which share the other conditions, even if that precedent is not explicitly recalled. At the same time, if the precedent is directly cited in the case, the system is capable of putting a stronger accent on that interpretation, not only by assuming its applicability but by directly stating it.

5 Carneades application

5.1. Reconstructing Precedents and their reasoning

Let's suppose we have a contract clause which - we are afraid – is oppressive. We start by querying the system (the ontologies and the rules) to retrieve a list of contract clauses (object of precedents) which have been considered oppressive – either as an undisputed fact or following a judicial interpretation (fig. 3).

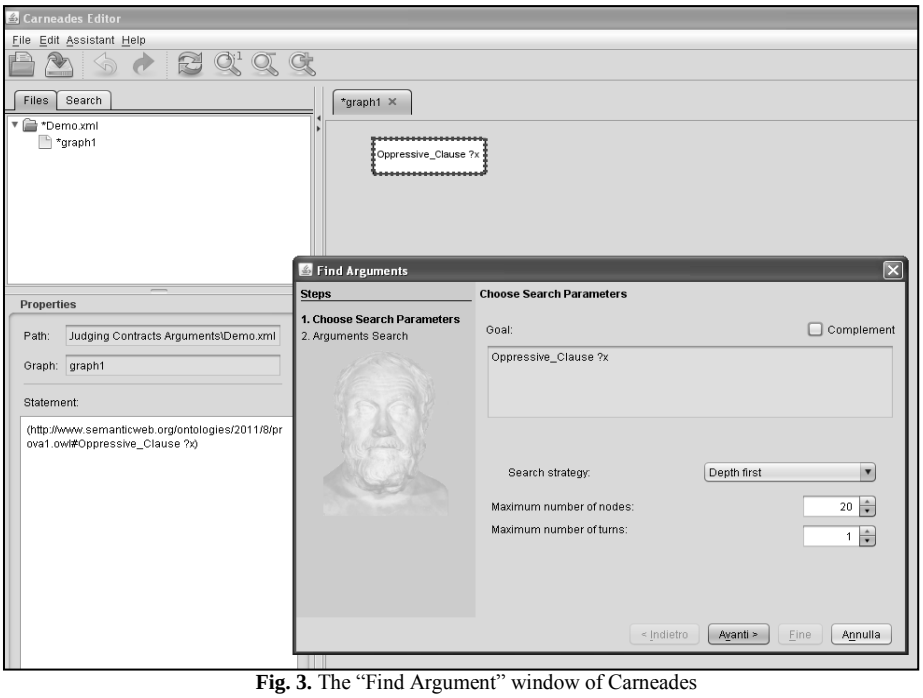


Fig. 3. The "Find Argument" window of Carneades

Fig. 4 shows the resulting list.

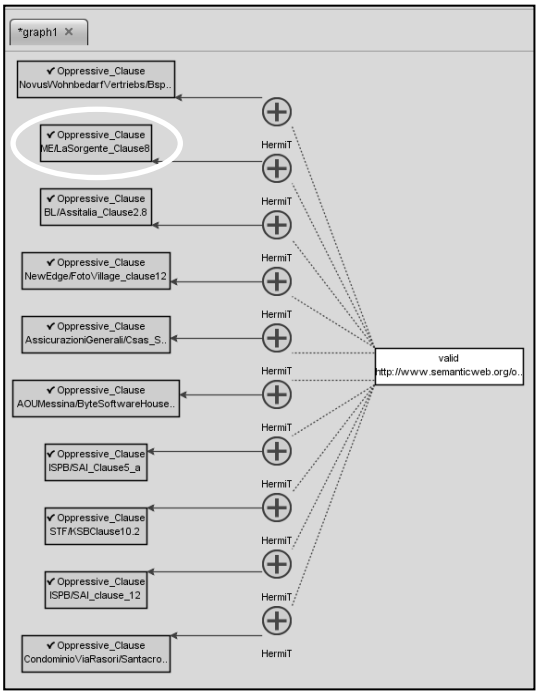


Fig. 4. The query results

We take the second example: ME/LaSorgente_Clause8.
Please notice that this instance does not represent a legal case, but rather a contract clause. Information contained in the ontology allows us to retrieve in any moment the details concerning the case (court, date, parties, the decision's text, and so on) but no such information will be shown on the argumentation tree: in it, we will only find the contract clause and other material circumstances related to it (the contract it is contained in, for example), legal statuses, and interpretation instances: only the elements which are relevant to the dynamics of the judicial argumentation, while those who pertain to the identification of the case are retrievable but not shown here.

We ask the system to tell us if ME/LaSorgente_Clause8 can be considered "inefficacious", in the light of applicable laws and judicial interpretations made by the judge in the precedent, and analyze the result (fig. 5).

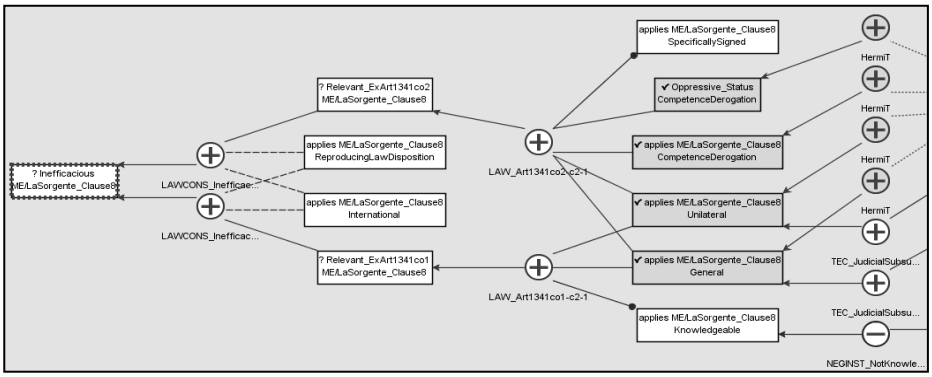


Fig. 5. Argumentation PRO the contract clause being inefficacious.

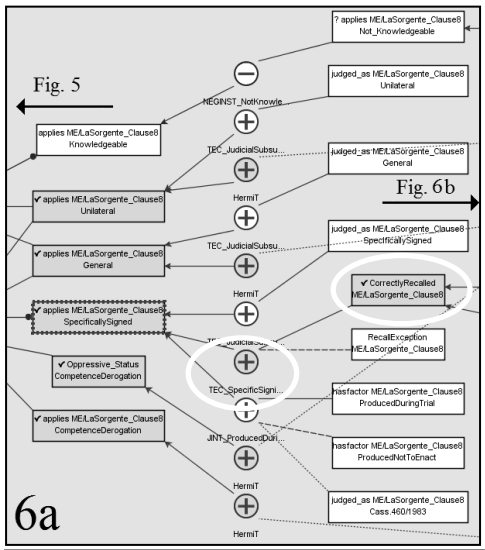
The system found two applicable laws to argument the inefficacy of the clause (left part of fig. 5): if the conditions of one of these two laws are met, and no exception exists (in this case, possible exceptions - broken lines - are the contract being an

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 fig.5.

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 (please notice that, due to a mistake, knowable appears as "knowledgeable" in the figures).

In order for the clause to be relevant under Article 1341co2, it must be general, unilateral, oppressive and not specifically signed.

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). These statements come directly from the ontology set's knowledge base, 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 "HermiT" argument followed by dotted lines pointing to the premise "valid [ontologyURI]"). The search was not deep enough to determine whether this clause is specifically signed or not, nor whether it was knowable or not (white boxes).



The next step is to ask Carneades to produce argumentation towards the acceptance/rejection of the yet undecided statements. We start asking to find arguments PRO the clause being specifically signed (Fig. 6):

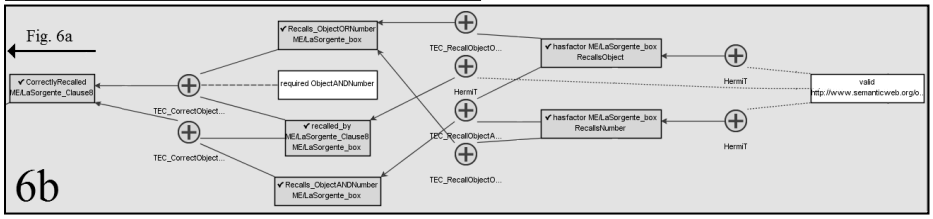


Fig. 6 (a-b). Argumentation PRO the clause being specifically signed.

Fig. 6a 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).

Fig. 6b 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. We do not know if the indication of both object and number in the distinct box is required for the recall to be considered as sufficient in the present case (it depends on the precedents that the judge decides to follow) but it doesn't matter, as long as our case fulfils the most restrictive requirement of indicating both the object and the number.

So we found a position to support the specific signing of the clause (the argument being called "Specific Signing Through Recall"), but – as we can see – it is prone to an exception which has not yet been explicitly rejected. Before checking it, we want to see if there is some plausible argumentation path leading to the opposite conclusion (which means, CON the clause being specifically signed):

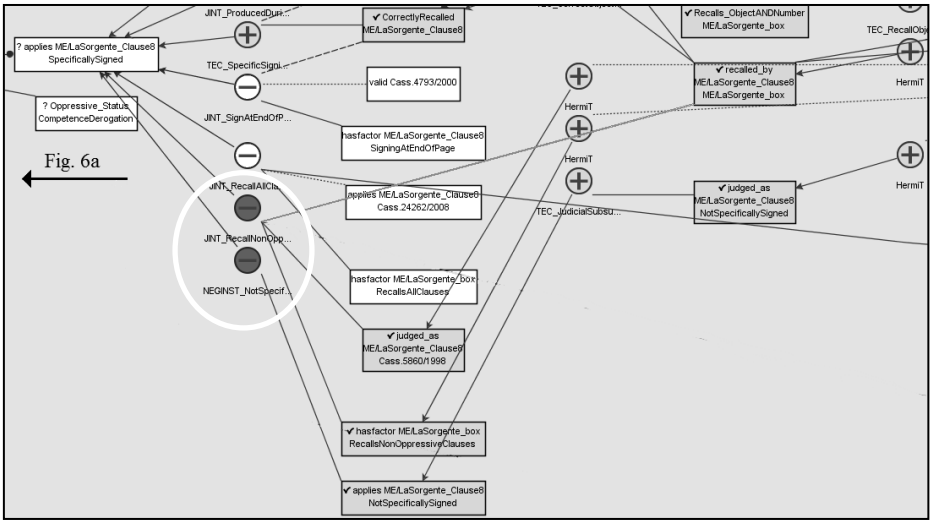


Fig. 7. Argumentation CON the clause being specifically signed.

In Fig. 7 we see that Carneades found two paths leading to the opposite conclusion, 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 DL-, 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, we see that the statement comes directly from the ontology set's knowledge base (ME/LaSorgente_Clause8 judged_as NotSpecificallySigned): it is a judicial interpretation made by the judge on that very clause). So, we now know that the judge actually interpreted that clause as not having been specifically signed – But why? To answer that question, we must look at the second argument (JINT_RecallNonOppressiveClause): we realize that the box which contains the specific signing contained 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.

This suggests 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 we have to proceed with one last step: checking if the exception for the PRO argument towards the specific signing of the clause applies (Fig. 8).

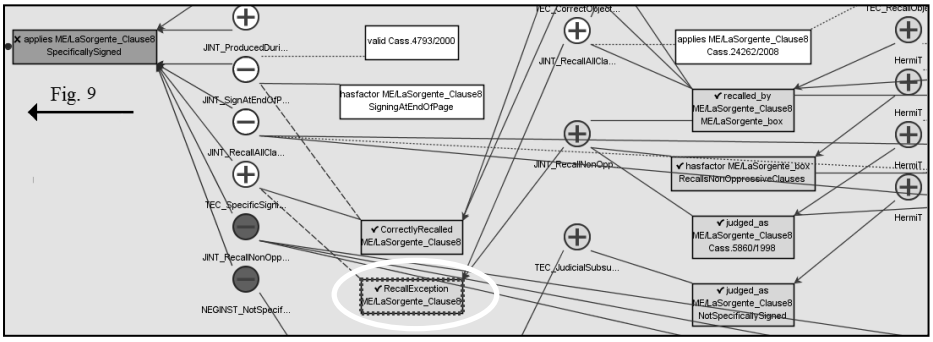


Fig. 8. Argumentation on the acceptation of the exception.

The argumentation graph of Fig. 5 is now complete: the oppressive clause has been judged as inefficacious, and the graph in Fig. 9shows why - and how it could be that a similar case gets a different outcome (white arguments):

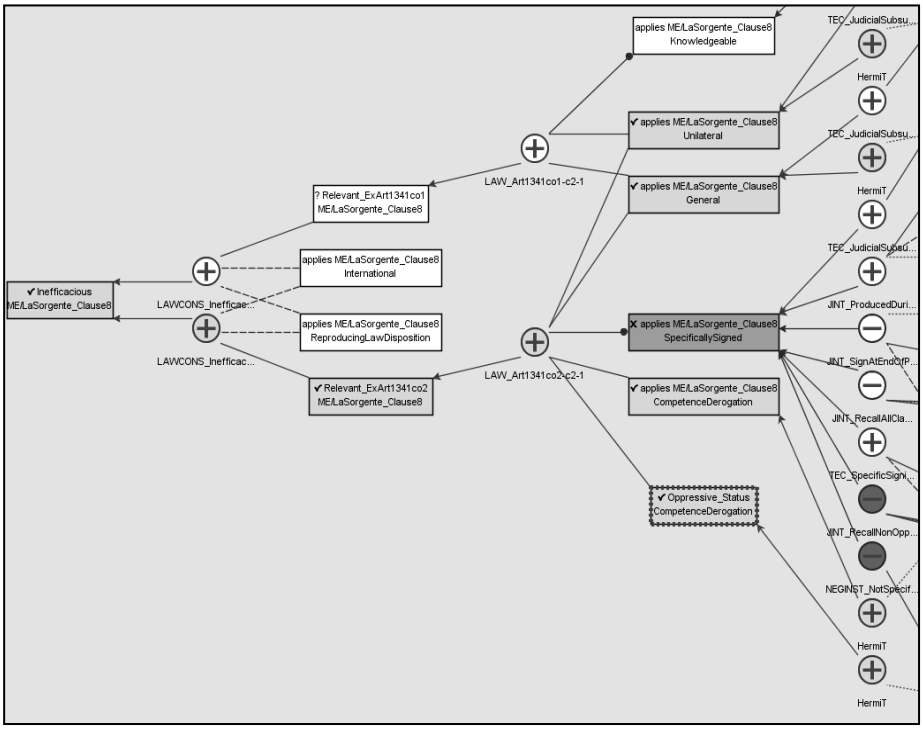


Fig. 9. The complete argumentation graph

5.2. Suggesting New Argumentation Paths

Finally, we provide a brief example of how Carneades can suggest new interpretation for known facts, in the light of existing norms and relevant precedents. We consider the same clause (ME/LaSorgente_Clause8) and ask the system to go through the analysis of its relevance under article 1342co2 (see fig. 1). Under that perspective, the clause lacked an acceptable argument PRO or CON its knowability. We try to find some arguments PRO the knowability and find 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). 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 (since there is no judged_as property 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.

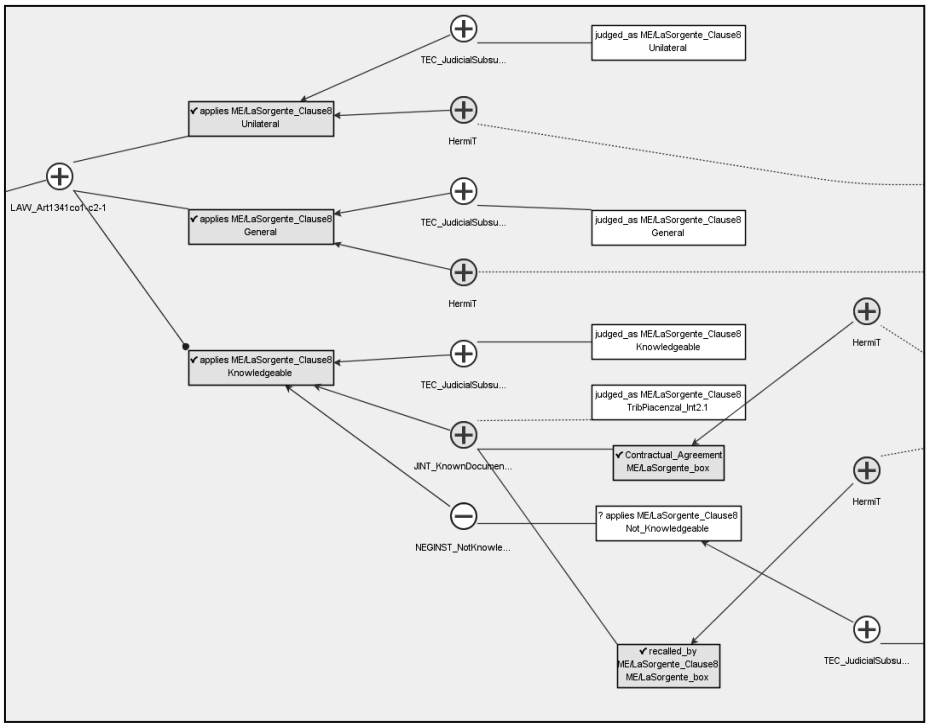


Fig. 10. Suggested argumentation PRO the clause being knowledgeable.

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.

6 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 query about 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). Finally, the defeasible logics behind the reasoner and the solid proof standards system - if properly implemented in the rule engine, and accompanied by an improvement in the reasoner's computability - can allow a complete analysis of possible exceptions to rules and interpretations, through a real “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.

7 Conclusions

The Carneades application presented here 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 section 2.
The present application therefore represents 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 work presented at the RuleML Challenge 2011 [6] because it creates a complete juridical environment in order to achieve 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 modelled 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.

References

1. Boella G., Governatori G., Rotolo A., van der Torre L., A Logical Understanding of Legal Interpretation. In: KR 2010.
2. Boer, A., Radboud, W., 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, pp. 21-41 (2008).
3. Ceci M., Combining Ontology and Rules to Model Judicial Interpretation, RuleML 2012 Doctoral Consortium (under submission).
4. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. In: Artificial Intelligence, 77(2), pp. 321-357 (1995).
5. Gordon, T.: Visualizing Carneades argument graphs. In: Law, Probability and Risk, 6(1-4), pp. 109–117 (2007).
6. Gordon, T.: Combining Rules and Ontologies with Carneades. In: Proceedings of the 5th International RuleML2011@BRF Challenge, CEUR Workshop Proceedings, pp. 103-110 (2011).
7. Gordon, T.: Constructing Legal Arguments with Rules in the Legal Knowledge Interchange Format (LKIF). In: Casanovas, P.: Computable models of the law: Languages, dialogues, games, ontologies. Berlin: Springer. (Lecture Notes in Artificial Intelligence 4884), pp. 162-184 (2008).
8. Gordon, T., 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, BERLIN, Springer pp. 282 - 296 (2009).
9. Gordon, T., Prakken, H., Walton, D.: The Carneades model of argument and burden of proof. In: Artificial Intelligence, Vol.171, No.10-15, pp. 875-896 (2007).
10. Gordon, T., Walton, D.: The Carneades Argumentation Framework: using presumptions and exceptions to model critical questions. In: Dunne, P.E.: Computational Models of Argument. Proceedings of COMMA 2006: 1st International Conference on Computational Models of Argument, The University of Liverpool, UK, 11th - 12th September 2006. IOS Press, Amsterdam (2006).
11. Gordon, T., Walton, D.: Legal reasoning with argumentation schemes. In: Proceedings of the Twelfth International Conference on Artificial Intelligence and Law, ACM Press, New York, pp. 137-146 (2009).
12. Palmirani, M., Ceci, M.: Ontology framework for judgement modelling, AICOL 2011.
13. Palmirani, M., Contissa, G., Rubino, R.: Fill the Gap in the Legal Knowledge Modelling. In: Proceedings of RuleML 2009, pp. 305-314 (2009).
14. Walton, D.: The New Dialectic: Conversational Contexts of Argument. University of Toronto Press, Toronto/Buffalo (1998).