Counterfactuals and Hybrid Reasoning in an Ontology of Law Articles

El Hassan Bezzazi

IREENAT, University of Lille 2, bezzazi@univ-lille2.fr

Abstract. In this paper, we present a formal legal cybercrime ontology using concrete tools. The purpose is to show how law articles and legal cases could be formally defined so that the problem of case resolution is reduced to a classification problem as long as cases are seen as subclasses of articles. Secondly, we show how counterfactual reasoning may be held over it within the framework of Description Logic. Lastly, we investigate the implementation of a hybrid system which is based both on this ontology and on a non-monotonic rule based system which is used to execute an external ontology dealing with a technical domain in order to clarify some of the technical concepts.

1 Introduction

We investigate in this paper the implementation of a formal ontology for criminal law dealing with cybercrime which is both functional and applicative. Our objective is twofold. First, we wish to present the ontology and an example of counterfactual reasoning it would use for its definition in a less abstract way than usual by using concrete tools. We use Protégé (Protégé 2007) which is an ontology editor supporting the OWL language and Racer (Racer 2007) which is a reasoning system based on Description Logics. We wish also to depict in the case of an interdisciplinary collaboration the clarification of some technical concepts through the use of a non-monotonic inference engine. This clarification brings to the ontology new information that may have consequences in the judge decision. Having worked on law texts related to computer security, we have chosen cybercrime as a subfield of criminal law as long as it constitutes a relatively small closed field. This paper is structured as follows. In the following section we remind some of the basic ideas related to formal ontologies, typically Protégé ontologies, and to Description Logics. In section 3, the corpus of interest is described and structured into classes. Section 4 is devoted to the reasoning mechanism that allows us solve a case by identifying the law articles covering it. Issues related to concept fitting are pointed out in section 5 and a technique to achieve such an operation is presented in section 6.

2 Formal Ontologies

The use of ontologies in legal domains is an issue which has been intensively investigated (Asaro et al. 2003; Bench-Capon and Visser 1997; Breuker et al. 2002; Valente 1995). A formal ontology describes the concepts and the relations relating them in a given domain. The relations define the semantics. Building a formal ontology is especially recommended for domains expressed in natural language as documents and corpora. An immediate benefit from the definition of such a formal ontology is the normalization of the semantics materialized by a structured terminology. This normalization is most relevant in the case of an interdisciplinary collaboration where a given term may carry real ambiguity according to one field or another. Indeed, natural language is characterized by its contextual nature which may lead to different interpretations. Think, in a forensic context, of how computer data suppression might be understood by a judge with no special knowledge in computer science. Expressing the concepts in a formal language such as OWL helps stabilizing the interpretation of these terms. Besides, expressing a formal ontology in OWL makes it machine consumable. In computer science, at least three kinds of ontologies are to be distinguished (Sowa 2007). Terminological ontologies in which concepts are named and are structured using mainly relations of the sub-type/super-type kind. As a matter of fact, such an ontology which is sometimes referred to as taxonomy can be expressed by using rules as we will further do it in the case of the ontology of computer data suppression. Ontologies of the second kind are those of which the concepts are built by enumerating the instances which compose them on the basis of some metric which defines their similarity. These concepts come usually as a result of a classification and are not named beforehand. The third type of ontology is the most sophisticated. The concepts are defined by axioms generally expressed in a decidable fragment of first order logic, namely Description Logic. Logical inferences can then be implemented for the classification of new instances. Incontestably, Description Logic is currently the standard for expressing formal ontologies on the basis of the OWL language for example. Efforts are carried out to extend it to a system able to handle knowledge expressed in the form of rules. This way, requests could be sent to existing rule bases within the semantic Web (Eiter et al. 2004). Another advantage that we outline in this extension is the possibility of supplementing a knowledge representation based on Description Logic by a rule based representation when this is more adequate. The use of rules is all the more relevant when it comes to take into account certain exceptions which characterize non-monotonic reasoning.

2.1 Classes and Properties

Classes are concrete representation for concepts. Different classes may be identified for representing a given domain knowledge. They must afterwards be structured by linking them with relations. These can be subsumption relations or Protégé-OWL relations called properties. Properties are relationships between individuals and an inverse property may be defined for a given property. Classes are interpreted as sets of individuals of similar structure. Classes can be organized in subclass-superclass hierarchy. The graphical representation of a hierarchy uses nodes for concepts and arcs for subsumption relations. Concretely, a class is defined by describing the conditions to be satisfied by individuals for they belong to the class. Note that classes may overlap and can be made explicitly distinct.

3.2 Description Logics

A knowledge base using description logic as a knowledge representation tool has two components :
- the TBox which contains the terminology of the domain of interest.
- the ABox which contains assertions on individuals named through the defined terminology.

The vocabulary is composed by concepts which denote sets of individuals and roles which denote binary relations between individuals. The description language which is specific to each Description Logic system has a well defined semantic: each TBox or ABox declaration may be identified to a formula of first order logic or a slight extension of it. Description logic provides also reasoning tools to decide for example if a description is consistent or not or if it is more general than another. Elementary descriptions are atomic concepts and atomic roles. These allow more complex descriptions to be built with concept constructors. The Description Logic language we shall use is defined by the following assertions where C and D are concepts, A an atomic concept and R a role.

A	(atomic concept)
T et $\bot$	(universal concept and empty concept)
$\neg C$	(concept negation)
$C \cap D$ et $C \cup D$	(concept intersection and union)
$\forall R.C$ et $\exists R.C$	(value restriction and limited existential quantification)

A formal ontology is defined by a set of structured concepts and a number of inclusions between these concepts. The semantics of the concepts and roles is defined with respect to a domain of interpretation O which defines the interpretation of each constant A: $\iota(A)=a$. Concepts are interpreted as subsets of O and roles are interpreted as binary relations over O satisfying :

$\iota(T)=O$, $\iota(\perp)=\emptyset$
$\iota(\neg C)=O-\iota(C)$
$\iota(C\cap D)=\iota(C)\cap\iota(D)$, $\iota(C\cup D)= \iota(C)\cup\iota(D)$
$\iota(\forall R.C)=\{d\in O|(d,e)\in \iota(R)\Rightarrow e\in\iota(C)$ for all e in $O\}$
$\iota(\exists R.T)= \{d\in O|$ there exists e in O s.t. $(d,e)\in \iota(R)$ and $e\in\iota(C)\}$

Two frameworks are mainly referred to in practical logics: logic programming and first order logic. An important difference between these two frameworks is the close world assumption (CWA) admitted in the former and the open world assumption (OWA) admitted in the latter. Even if OWL admits primarily the OWA, CWA may be admitted if stated explicitly. CWA is very useful for dealing for example with the application of forward chaining. If in a rule base, only the rule "IF offence OR crime THEN infringement" infers the fact infringement, CWA allows inferring that there is no infringement if none of the facts offence or crime is established.

4 The Corpus

We list in this subsection the French criminal law articles that are of interest to us and from which irrelevant metadata has been removed (Légifrance 2007).

Article 323-1: *Fraudulently accessing or remaining within all or part of an automated data processing system is punished by one year's imprisonment and a fine of € 15,000. Where this behaviour causes the suppression or modification of data contained in that system, or any alteration of the functioning of that system, the sentence is two years' imprisonment and a fine of € 30,000.*

Article 323-2: *Obstruction or interference with the functioning of an automated data processing system is punished by three years' imprisonment and a fine of € 45,000.*

Article 323-3: *The fraudulent introduction of data into an automated data processing system or the fraudulent suppression or modification of the data that it contains is punished by three years' imprisonment and a fine of € 45,000.*

Article 323-4: *The participation in a group or conspiracy established with a view to the preparation of one or more offences set out under articles 323-1 to 323-3, and demonstrated by one or more material actions, is punished by the penalties prescribed for offence in preparation or the one that carries the heaviest penalty.*

We shall consider in what follows three concepts (Figs. 1–3): Malicious actions which are punished by criminal law, responsibilities related to an action and the criminal law articles. Other classes of our ontology such as Sanction and Infringement are of less interest in what we shall expose.

Several actions may be qualified as being malicious in computer security and put in classes like privacy or hacking which in its turn covers classes like intrusion, denial of service….etc.

The class Malicious_Act depicts a classification for a sample of malicious actions:

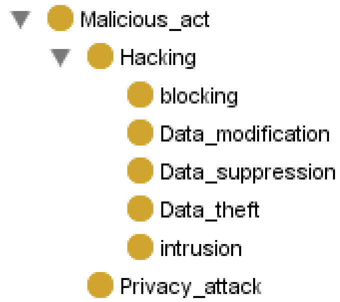

Fig. 1. The class of malicious actions

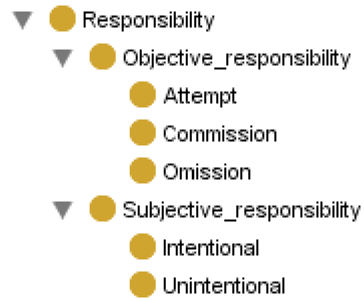

Fig. 2. the class of responsibilities

Fig. 3. The class of articles

Criminal law makes a distinction between two types of responsibilities, objective responsibility which may be commission, omission or attempt and the subjective responsibility which describes the intentional nature of the action. Of course the classes and subclasses defining these concepts are exclusives.

Criminal law articles which are of interest to us are grouped in the class Articles. As a matter of fact there are so far only six articles that deal directly with cyber-crime. Among these, for our purpose, we shall consider in particular four articles. It should be outlined that the conception that we make of a law article makes of it a class which groups all the cases it allows to characterize, that is to say the cases which fall under this article!. In this respect, our model is different from (Asaro et al. 2003). The rationale behind this conceptualization is that the concept of case inherits of the same characteristics and properties as in the concept of article. Henceforth, the application of our ontology consists in classifying, if possible, each case of interest as a subclass of one or more subclasses of Articles.

Listed below in Fig. 4 are some of the relations of interest we shall use here. In particular, between the two classes Articles and Responsibility the relation hasResponsibility specifies the nature of the responsibility handled in the article which may be commission, omission or attempt in the case of objective responsibility or which may be intentional or unintentional in the case of subjective responsibility. For each relation its inverse relation is given. Inverse relations are very useful and enhance the way of expressing axioms as we shall see below.

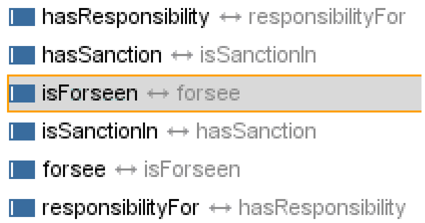

Fig. 4. Ontology properties

4 Using A Reasoner

The possibility of using a reasoner to infer automatically the hierarchy of classes is one of the major advantages in using OWL-DL. Indeed in the case of important ontologies containing hundreds of classes the use of a reasoner is crucial, in particular when dealing with multiple inheritance. Thus the designer will focus on logical description which is hierarchical, flexible and consequently easy to maintain.

4.1 Articles Conceptualization

Ontologies which are described in OWL-DL may be processed by a reasoner. One of the main tasks handled by a reasoner is to check if a given class is a subclass of another class. Another task is to check consistency, the reasoner can check on the basis of the class conditions if the class may have instances or not. A class which has no instances is inconsistent. Thus a class which is defined to be a subclass of both classes A and B which are disjoint will be detected as inconsistent by the reasoner. Necessary conditions are used to express « if an object is in this class it necessarily must satisfy these conditions ». A class which uses only necessary conditions is called partial. Necessary and sufficient conditions are used to express « if an object is in this class it necessarily must satisfy these conditions and if an individual satisfy these conditions then it necessary belongs to this class». Such a class is said to be complete and allow a CAW reasoning. All the classes we shall deal with in this paper are complete. The axioms defining the classes Art_323-1, Art_323-2, Art_323-3 and Art_323-4 are shown in Protégé screenshots given in Figs. 5–8.

Several actions may be qualified as being malicious in computer security and put in classes like privacy or hacking which in its turn covers classes like intrusion, denial of service….etc. The class Malicious_Act depicts a classification for a sample of malicious actions. Criminal law makes a distinction between two types of responsibilities, objective responsibility which may be commission, omission or attempt and the subjective responsibility which describes the intentional nature of the action.

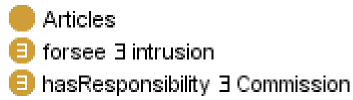

Fig. 5. The class Art323-1

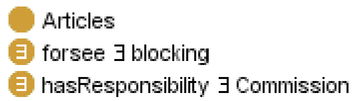

Fig. 6. The class Art323-2

Fig. 7. The class Art323-3

Fig. 8. The class Art323-4

4.2 Reasoning with Counterfactuals

We are going to depict the expressive power of Description Logics through an example where it is made an assumption that contradicts the reality. This kind of reasoning is called counterfactual reasoning (Ginsberg 1986). It allows reasoning on abstract facts which are inconsistent with actual facts. From the point of view of logical semantics, a counterfactual is always true. However in commonsense reasoning the truth value assigned to a counterfactual depends on the meaning it carries and which depends on the current context. In (Ginsberg 1986) the author gives an account of counterfactuals in terms of possible worlds initiated by (Lewis 1973) and identifies some of the domains where counterfactuals have application such as planning, diagnosis and natural language understanding. In this paper, we are considering this form of reasoning when writing and using law articles while thinking at the same time to cases to which these articles do apply. Typically, in criminal law, an article describes the conditions which, if satisfied by a given case, a sanction is entailed. It may also happen that the definition of another article is made by reference to other articles possibly by relaxing or even contradicting some of their conditions. We shall consider in what follows an approach we propose for representing these mental constructions in Description Logics. It is based on the fact that in propositional logics the for-

mula $X \Rightarrow Y$ is equivalent to $\neg X \vee Y$ on one hand and on the fact that proving a disjunction comes to prove one of its disjuncts. Let B and D be the conditions that characterize a case falling under article A, $A=B\cap D$. On the other hand, article A' is literally defined by requiring instead of B a condition B' and keeping D. B' may contradict B. Of course we could simply write $A'= B'\cap D$ but what we wish to do is to stick as much as possible to the literal expression of the article A' which would not mention D but mentions A as done in Art_323-4. We write $A'=B'\cap (B\Rightarrow A)$ to express that A' applies when the case is characterized by B' and if we suppose that B holds (even if it does actually not) the case falls under A. Therefore, we have:
$A'=B'\cap(\neg B\cup A)=B'\cap(\neg B\cup B\cap D)=B'\cap(\neg B\cup D)=B'\cap\neg B\cup B'\cap D$
To prove that a given case C falls under A', one should prove that either $C\subseteq B'\cap\neg B$ or $C\subseteq B'\cap D$. If the ontology definition is made in such a way that it is not possible for the reasoner to infer $C\subseteq B'\cap\neg B$ then the only way to prove $C\subseteq A'$ through the definition of A' is by proving $C\subseteq B'\cap D$.
The key point here is that we have succeeded this way to express A', $A'=B'\cap (B\Rightarrow A)$, in a manner which is close to its textual definition by not mentioning D while proving that a case C falls under A' comes to proving that $C\subseteq B'\cap D$.

For example, solving a case which falls under article Art_323-4 needs as stated by this even article to compare the case to articles Art_323-1, Art_323-2 and Art_323-3. Solving the case is made possible by making an assumption in the definition of Art_323-4 which is contrary to what is stated in it. Indeed, think of a case defined by Attempt and Intrusion. To realize that this case falls under article_323-4, one should first assume that in case the responsibility was Commission then the case would have fallen under Article_323-1. This is a counterfactual reasoning as long as the assumption Commission is contrary to the Attempt responsibility which characterizes the case at hand.
∃forsee.∃isForseen.(¬(∃hasResponsibility.Commission)∪(Art_323-1∪Art_323-2∪Art_323-3)))

According to the interpretation rules given above, this is to be understood as the class of articles that foresee malicious actions that are foreseen in articles Art_323-1, Art_323-2 or Art_323-3, by assuming Commission responsibility. Rewritten as: $\exists f.\exists i.(\neg C\cup(A1\cup A2\cup A3))$

A1, A2 and A3 are the axioms defining the three first articles. A and C stand respectively for articles stating Attempt and Commission responsibility. To isolate within the articles the stated malicious actions from the responsibility, axioms A1, A2, A3 are rewritten as:

$A1\equiv C\cap D1$
$A2\equiv C\cap D2$
$A3\equiv C\cap D3$

On the other hand:

$A4= C\cap D4$ with

$D4=\exists f.\exists i.(\neg C\cup(C\cap D1\cup C\cap D2\cup C\cap D3))$

Thus :

$\exists f.\exists i.(\neg C \cup (C \cap (D1 \cup D2 \cup D3)))$

It is easy to prove in propositional logic :

$\neg C \cup (C \cap X) = \neg C \cup (\neg C \cap X) \cup (C \cap X) = \neg C \cup ((\neg C \cup C) \cap X) = \neg C \cup X$

Therefore, we have:

$\exists f.\exists i.(\neg C \cup D1 \cup D2 \cup D3)$

Note, as we have seen above, that we have managed this way to evacuate from Art_323-1, Art_323-2 and Art_323-3 the Commission responsibility to make things consistent. Fig. 9 shows the resolution of three cases. Case_1 consists in both system blocking and data modification which have been committed, thanks to multiple inheritance, and case_2 is a case where an intrusion attempt has been stated. Case_3 is an example of case that might not be resolved, for example a case referring to data theft which does not appear explicitly in the corpus.

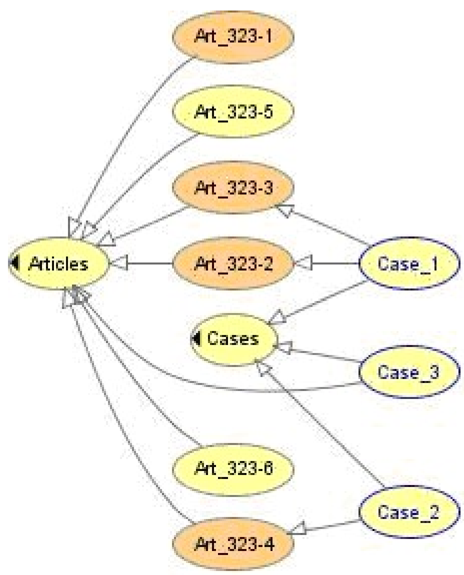

Fig. 9. The inferred hierarchy

5 Fitting Technical Concepts and Legal Concepts

Mismatching between legal concepts and technical concepts constitutes a serious issue (De Lamberterie and Videau 2006). In (van Laarschot et al. 2005) the authors address a similar topic where the issue is about making layman's terminology fit a legal terminology when describing their case. In our case, consider for example computer data suppression happens to be mentioned in some of criminal law articles. With no explicit legal definition, this naturally leads the judge to adopt the natural language definition for suppression. The common understanding of the term suppression is physical suppression where a thing which is suppressed merely stops existing. However, in the computer world, suppressing data means very often logical suppression where data could be restored with adequate tools. In addition to that, even in the case of a physical suppression, computer data could be restored when a backup or archiving politic is observed by the data processor. This semantic difference should be definitely specified because the penal consequences for fraudulent computer data suppression may vary according to the possibility of recovering the data. This means that although the action is condemnable in both cases, the sanction might be worsened or attenuated depending on the type of suppression. To make the common understanding of the term suppression fit the effective definition of the term computer data suppression, one solution consists in « connecting » its concept in a legal ontology to its concept in a computer ontology. This connection may need some new concepts and new relations be added to the two already existing ontologies. New concepts may also be needed to summarize or to extract from the second ontology that information which is readily of interest for a legal reasoning. For example, in the case we are dealing with, such new concepts are « restorable data» and «unrestorable data». These ontological adjustments may prove to be disproportionate in case where the relevant information is well defined. It is indeed sufficient to compute this information by using a rule based inference engine. As a matter of fact, the second ontology is principally used to deduce facts rather than for classification.

6 Hybrid Reasoning

In a case where only the subsumption relation is used to deduce relevant facts, it is sufficient to use the second ontology in a rule based form within a propositional logic framework (Fig. 10). However the inference engine to be used should allow non-monotonic reasoning if we wish the ontology preserve its structure in this translation and in the same time manage conflicting facts. The use of defeasible reasoning in ontology integration is depicted through scenarios in (Bassiliades et al. 2006). We have chosen to use an inference engine based on stratified forward chaining which, through an adequate backward chaining (Bezzazi 2006) sends questions to the user to compute which of the facts « existing data » or « non-existing data » holds for the suppressed data. It should be noticed that the concept of legal suppression as well as the concept of computer data suppression, both inherits somehow of the French language concept of the term suppression which normally entails the no more existence of the suppressed object. Indeed, according to the French definition, to suppress something is to be understood as putting an end to the existence of something.

```
French_suppression > !existence
```

```
Legal_suppression > French_suppression
data_suppression  > French_suppression.
```

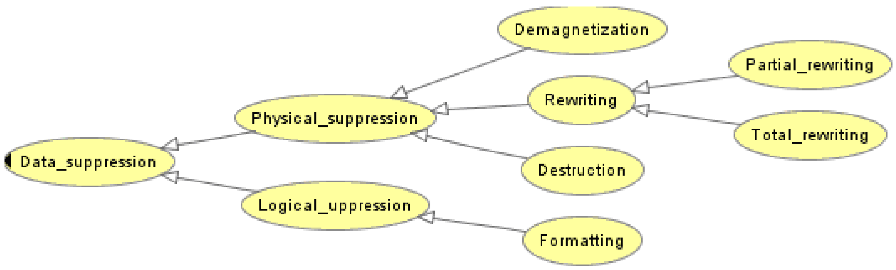

Fig. 10. The taxonomy of data suppression

The translation of this ontology fragment as a rule base yields:

```
Logical_suppression > data_suppression
Physical_suppression > data_suppression
Formatting > Logical_suppression
Destruction > Physical_suppression
Demagnetization > Physical_suppression
Rewriting > Physical_suppression
Partial_ rewriting > Rewriting
Total_rewriting > Rewriting
```

We add a rule which expresses that data which has been logically suppressed may still exist.

```
Logical_suppression > existence
```

If logical suppression is established, the stratified forward chaining will, like an inheritance system with exceptions, give priority to the application of this last rule with respect to the more general rule:

```
French_suppression > !existence
```

Therefore, this rule base should help the lawyer or the judge make their decisions or instruct a case by shedding light on a technical concept lacking a legal definition. The explanation process is done through a question-response procedure.

7 Conclusion

The framework we have presented in this paper is based on the idea of considering cases as being, by their structure, subclasses of articles. Therefore, the problem of solving a case is the same as that of classifying it. With such a system at work, all one has to do is to implement articles as classes which should not be a difficult task at least manually. Doing this in a semi automatic or automatic way constitutes an interesting topic for investigation. On the other hand, we have shown, in a rather practical way, how counterfactual reasoning and non-monotonic reasoning which are naturally used in legal reasoning may be performed in our system. As a matter of fact, the cybercrime ontology is implemented, the non-monotonic inference engine is also implemented but interfacing these to achieve the so-called hybrid reasoning is only projected and further work need to be done on this topic.

References

Asaro, C., Biasiotti, M.A., Guidotti, P., Papini, M., Sagri, M.T., Tiscornia, D. A, (2003) Domain Ontology: Italian Crime Ontology, ICAIL 2003 Workshop on Legal Ontologies

Bassiliades, N., Antoniou, G. and Vlahavas, I. (2006) A Defeasible Logic Reasoner for the Semantic Web, 2006, International Journal on Semantic Web and Information Systems 2,1

Bench-Capon, T.J.M., and Visser, P.R.S., (1997) Ontologies in Legal Information Systems: The Need for Explicit Specifications of Domain Conceptualisations. In Proceedings of the Sixth International Conference on AI and Law. ACM Press.

Bezzazi, H. (2006). On some inferences based on stratified forward chaining: an application to e-Government. Proceedings of the 15th International Conference on Information System Development, Springer Verlag.

Breuker, J., Elhag, L., Petkov, E. and Winkels,R. (2002). Ontologies for legal information serving and knowledge management. In Legal Knowledge and Information Systems. Jurix 2002: The Fifteenth Annual Conference. Amsterdam. IOS Press.

De Lamberterie, I. and Videau, M. (2006). Regards croisés de juristes et d'informaticiens sur la sécurité informatique. In Symposium sur la Sécurité des Technologies de l'Information et des Communications. Rennes, mai-juin.

Eiter, T., Lukasiewicz, T., Schindlauer, R. and Tompits. H. (2004) Combining answer set programming with description logics for the semantic web. Proceedings of the 9th Int. Conf. on the Principles of Knowledge Representation and Reasoning.

Ginsberg, M. L. (1986) Counterfactuals. Artificial Intelligence 30:35-79.

Légifrance, Retrieved February 4, 2007, from Web site http://www.legifrance.gouv.fr/html/codes_traduits/code_penal_textan.htm

Lewis, D. (1973) Counterfactuals, Harvard University Press. Cambridge.

Protégé, Retrieved February 4, 2007, from Web site http://protege.stanford.edu

Racer, Retrieved February 4, 2007, from Web site http://www.racer-systems.com

Sowa, J. F. Ontology. Retrieved February 4, 2007, from Web site: http://www.jfsowa.com/ontology

Valente, A. (1995) Legal Knowledge Engineering; A Modelling Approach, IOS Press, Amsterdam, The Netherlands.

van Laarschot, R., van Steenbergen, W., Stuckenschmidt, H., Lodder, A. and van Harmelen, F. (2005) The Legal Concepts and the Layman's Terms. Proceedings of the 18th International. JURIX conference, IOS Press.