Ontology Mapping for a Legal Question Answering System

Cassia TROJAHN [a,1], Paulo QUARESMA [a] and Renata VIEIRA [b]

[a] *Departamento de Informática, Universidade de Évora, Évora, Portugal*
[b] *Pos-Graduacão em Computacão Aplicada, Universidade do Vale do Rio dos Sinos, São Leopoldo, Brazil*

Abstract. Legal information retrieval systems, such question answering, use legal ontologies to represent semantic objects, to associate them with legal documents and to make inferences about them. The ontology mapping process can help users to reuse and compare information from different ontologies. In this paper we present a review on legal ontologies and present an approach to ontology mapping based on argumentation. Individual mappings are computed by specialized agents using different mapping approaches. Next, these agents use argumentation to exchange their local results, in order to agree on the obtained mappings. To each argument is associated a *strength*, representing how confident an agent is in the similarity of two ontology terms. Based on their preferences and confidence of the arguments, the agents compute their preferred mapping sets. The arguments in such preferred sets are viewed as the set of globally acceptable arguments. This work is part of a question answering system for the legal domain.

Keywords. Ontology mapping, legal ontologies, argumentation framework

Introduction

Legal ontologies provide a formal description of the objects and their relations in the legal domain. Legal information retrieval systems, such as question answering systems, use this knowledge to represent semantic objects, to associate them with legal documents and to make inferences about them. Core legal ontologies covering different aspects of the legal domain have been proposed in the literature. The mapping process, takes two ontologies as input and determines as output correspondences between the semantically related entities of those ontologies, can help users to reuse and compare information one from the other. It is specially interesting when extending a core ontology (see [8]). In the legal domain core ontologies are often considered as a start point for ontology engineering (see, for example [5]).

In this paper we present a review on legal ontologies and present an approach based on argumentation to combine different techniques to ontology mapping. Different ontology mapping approaches are combined, as terms may be mapped by a measure of lexical similarity ([23][18]), or they can be evaluated semantically, usually on the basis of semantic oriented linguistic resources, or considering the term positions in the ontology

hierarchy ([13]). It is assumed that the approaches are complementary to each other and combining different ones reflect better solutions when compared to the solutions of the individual approaches.

First, individual mappings are computed by specialized agents using different mapping approaches (lexical, semantic and structural). Next, these agents use argumentation to exchange their local results, in order to agree on the obtained mappings. An Extended Value-based Argumentation Framework (E-VAF) is used to represent arguments with *strength* [24]. The E-VAF allows to determine which arguments are acceptable, with respect to the different *audiences* represented by different agents. To each argument is associated a *strength*, representing how confident an agent is in the similarity of two ontology terms. Based on their preferences and confidence of the arguments, the agents compute their preferred mapping sets. The arguments in such preferred sets are viewed as the set of globally acceptable arguments.

This paper is structured as follows. Section 2 present a review on legal ontologies. Section 3 presents our argumentation model. Section 4 contextualize our work in a question answering system for the legal domain. Section 5 presents a walk through example. Finally, section 6 presents the final remarks and the future work.

1. Legal Ontologies

In this section, we first comment on ontologies in general. Next, a review on ontologies for legal domain is presented. First of all, we classify ontologies according to the categories presented by [11]: *lightweight ontologies*, which consists of a set of hierarchically organized terms; *upper or top ontologies*, which attempts to describe fundamental categories applicable to all domains; *core or domain ontologies*, which attempts to articulate the concepts fundamental to some particular domain. and *application ontologies*, which contains the very detailed and specific concepts required to perform a particular task on a particular piece of law.

Four proposal of legal ontologies can be considered the first attempts to formalize the entities of the legal domain: Language for Legal Discourse (LLD) developed by [19] that addressed the problem of representing modalities, specially those related to *time*, *events*, *actions* and *deontic* notions of *permission* and *obligation*; the ontology basis described in [22], a formalism to be used to model social systems, such as law; the Frame Based Ontology of [1][27] that developed a generic ontology used by [27], who developed a specific ontology describing Dutch Unemployment Benefit Law; and Functional Ontology of Law (FOLaw) proposed by [26] which intended to be a top ontology to classify the various elements that are required to make up a functioning legal system.

More recent ontologies are LRI-Core (Real Core Ontology for Law) [6], which is intended to be a core ontology for legal domains, that identifies the main concepts that are typical, and preferably exclusive for law. The *Law in the Net* [12] is an example of a lightweight ontology which aims to define and promote a controlled language for legislation. Following the same idea of automated acquisition of ontologies, [20] propose a methodology for applying Natural Language Processing (NLP) techniques to automatically create a legal ontology from legal documents. [8] describes the construction of an application ontology around the concept of EMPLOYEE in a European community legislation text. An example of application domain ontology is the IPROnto [7], an ontology

of the Digital Rights Management (DRM) domain. In a similar domain, more recently, [2] describes a conversion of an expert system on the law governing the sale of goods, Article II of the Uniform Commercial Code (UCC), into a ontology.knowledge- based system using the Web Ontology Language OWL. Other recent proposal is the E-POWER project [4], a project undertaken for the Dutch Income Tax organization. The Core Legal Ontology (CLO) is used to support the construction of legal domain ontologies [10]. A current approach is LKIF (Legal Knowledge Interchange Format) [14], a legal core ontology that is part of a generic architecture for legal knowledge systems. [15] describe an ontology used as a common sense between the terminology that laymen use to describe their case and the terminology found in legal documents. Finally, [21] present a natural language biased top-level ontology extended with respect to the legal domain. NM-L is an extension of an NM core ontology which has no legal bias.

2. An Extended Value-Based Argumentation Framework

Our argumentation model is based on the Value-based Argumentation Framework (VAF)[3], a development of the classical argument system of Dung [9]. In Dung's frameworks, attacks always succeed. However, in many domains, including the one under consideration, arguments lack this coercive force: they provide reasons which may be more or less persuasive [16]. Moreover, their persuasiveness may vary according to their audience. The VAF is able to distinguish attacks from successful attacks, those which defeat the attacked argument, with respect to an ordering on the values that are associated with the arguments. It allows accommodating different audiences with different interests and preferences. We extend the VAF in order to represent arguments with *strength*, which represents the confidence that an individual agent has in some argument. Two elements have been added to the VAF: a set with *strength* and a function which maps from arguments to *strength*. We assumed that the *strength* is a relevant criteria to represent the ontology mapping domain.

In previous work [24] we had used only two discrete classes to express the *confidence degree* of an agent had in the mappings (*certainty* and *uncertainty*). In this paper, we propose a *strength* $\in [0,1]$ for the arguments, considering the similarity between the tokens of the terms, considering the length of the terms and the number of tokens that match to each other, as detailed in Section 5.

Definition 3.1 An Extended Value-based Argumentation Framework (E-VAF) is a 7-tuple *E-VAF = (AR, attacks,V,val,P,C,valC)* where *(AR,attacks,V,val, P)* is a value-based argumentation framework, *C* is a nonempty set of values representing the *strength*, *valC* is a function which maps from elements of *AR* to elements of *C*. *valC* $\subseteq C \times C$ and *valprefC(c_1,c_2)* means c_1 is preferred to c_2.

Definition 3.2 An argument $x \in AR$ defeats$_a$ (or *successful attacks*) an argument $y \in AR$ for audience *a* if and only if *attacks(x,y)* $\wedge$ *(valprefC(valC(x), valC(y))* $\vee$ *(¬ valpref(val(y),val(x))* $\wedge$ *¬ valprefC(valC(y), valC(x))))*.

An attack succeeds if (a) the *strength* of the attacking argument is greater than the *strength* of the argument being attacked; or if (b) the argument being attacked does not have greater preference value than attacking argument (or if both arguments relate to the

same preference values) and the *strength* of the argument being attacked is not greater than the attacking argument.

Definition 3.3 A set S of arguments is *conflict-free* for audience a if $(\forall x)(\forall y)$ $((x \in S$ & $y \in S)$ $\longrightarrow$ $(\neg attacks(x, y)$ $\vee$ $(\neg valprefC(valC(x),valC(y))$ $\wedge$ $(valpref(val(y), val(x))$ $\vee$ $valprefC(valC(y),valC(x)))))$.

3. E-VAF for DL Ontology Mapping

In our model, dedicated agents encapsulate different mapping approaches. Each approach represents a different audience in an E-VAF, i.e, the agents' preferences are based on specific approach used by the agent. In this paper we consider three audiences: lexical (L), semantic (S), and structural (E) (i.e. $P = \{L, S, E\}$, where $P \in$ E-VAF). We point out that our model is extensible to other audiences.

3.1. Argumentation generation

First, the agents work in an independent manner, applying the mapping approaches and generating mapping sets. The mapping result will consist of a set of all possible correspondences between terms of two Description Logic ontologies (OWL-DL, as commented in Section 5). A mapping m can be described as a 3-tuple $m = (t_1, t_2, h)$, where t_1 corresponds to a term in the ontology 1, t_2 corresponds to a term in the ontology 2, and h is one of $\{+,-\}$ depending on whether the argument is that m does or does not hold. Now, we can define arguments as follows:

Definition 4.1 An *argument* $\in AR$ is a 4-tuple $x = (m,a,s)$, where m is a mapping; $a \in P$ is the agent's audience generating that argument (agent's preference, i.e, lexical, semantic or structural); $s \in S$ is the *strength* of the argument. The *strength* is defined by the agent when applying the specific mapping approach. Here, we assumed $S \in [0,1]$, where $S \in$ E-VAF, as commented below.

3.1.1. Lexical agent

The lexical agent adopts a metric to compare string similarity. We used the *lexical similarity* proposed by [18]. This metric is based on the Levenshtein distance (edit distance) [17], which is given by the minimum number of operations (insertion, deletion, or substitution of a single character) needed to transform one string into another. The length of the compared terms is considered to compute the *lexical similarity*. This metric returns a value from the interval [0,1], where 1 indicates high similarity between two terms.

The first step in the mapping process is the tokenization. Terms are parsed into tokens by a tokenizer which removes stop words ("and", "of", etc). *strength* of an argument is computed according to the lexical similarity between each token of the two compared terms. The s corresponds to a value from the interval [0,1], which is computed according to the number of lexically similar tokens in the compared terms.

When the agent has *certainty* in the mapping (for instance, all terms are lexically similar to each other), the *strength* of the argument is 1. If some token of the terms are similar to each other, the *strength* is computed according to the number of similar tokens,

considering the length of these terms, as detailed below. Otherwise, if there is no similar tokens between the terms, the agent is not sure that the terms mapping (i.e., *strength* equals to 0), because this agent knows that other agent can be *certainty* in that mapping, with *strength* equals to 1. In the specific case of the lexical agent, the mapping can be resolved by the semantic agent, which can have an argument with *strength* equal to 1.

This way, if the lexical similarity is greater than a *threshold* and *all* tokens of the compared terms are lexically similar, the lexical agent generates an argument $x = (m,L,1)$, where $m = (t_1,t_2,+)$. Otherwise, if the lexical similarity is greater than a *threshold* and *some* tokens of the compared terms are lexically similar, the lexical agent generates an argument $x = (m,L,s)$, where $m = (t_1,t_2,+)$, where s is a value from the interval $[0,1]$, computed using the following formula, where T_S is the term from the source ontology, T_T is the term from the target ontology, and nM is the number of tokens that matches between t_S and t_T:

$$s := max\left(0, \frac{max(\mid T_S \mid, \mid T_T \mid) - (max(\mid T_S \mid, \mid T_S \mid) - nM)}{max(\mid T_S \mid, \mid T_S \mid)}\right)$$

Intuitively, this formula indicates that the greater the number of similar tokens t between T_S and T_T is the greater is the value of s. Finally, if there is no lexically similar tokens between the T_S and T_T, the agent generates an argument $x = (m,L,1)$, where $m = (t_1,t_2,-)$.

3.1.2. Semantic agent

The semantic agent consider the semantic (i.e., synonym, hyponym, and hypernym) relations between concepts to measure the similarity between them, on the basis of WordNet[2] database, a large repository of English semantically related items.

When the agent has *certainty* in the mapping (for instance, all terms have some semantic relation – synonymous, hypernym or hyponym – with each other), the *strength* of the argument is 1. If some tokens of the terms are similar to each other, the *strength* is computed according to the number of semantically related tokens, considering the length of these terms. Otherwise, if there is no similar tokens between the terms, the agent is not sure that the terms mapping (i.e., *strength* equals to 0), because this agent knows that other agent can be *certainty* in that mapping, with *strength* equal to 1. In the specific case of the semantic agent, when the searched terms is not available in the WordNet, the lexical agent can decide the mapping. It is common because there is no complete lexical database for every domain (i.e., the WordNet is incomplete for some domains).

This way, if the *all* tokens from the terms are *synonymous*, the value of s is 1, else it is computed according to the number of synonymous tokens. The *direct semantic relation* occurs when the Wordnet has some entry for the composed term, then s is 0 (the compared terms are related but not by *synonymous* relation). So, when the terms being mapped are *synonymous*, the agent generates an argument $x = (m,S,1)$, where $m = (t_1,t_2, +)$. The terms related by hyponym or hypernym are considered related and an argument $x = (m,S,s)$, where s can have value 1.0 (direct semantic relation) or a value according to the number of tokens semantically related from t_S and t_T.

---

[2]http://www.wordnet.princeton.edu

3.1.3. Structural agent

The structural agent considers the positions of the terms in the ontology hierarchy to verify if the terms can be mapped. First, it is verified if the super-classes of the compared terms are lexically similar. If not, the semantic similarity between they is used. If the super-classes of the terms are lexically or semantically similar, the terms can be matched. The argument is generated according to the lexical or semantic comparison. For instance, if the super-classes of the terms are not lexically similar, but they are synonymous (semantic similarity), an argument $x = (m,E,s)$, where $m = (t_1,t_2,+)$, is generated, where s varies according to the lexical or semantic analyze.

However, there are two main differences among the *strengths* returned by the structural, lexical and semantic agents. When the agents have not *certainty* in that mapping the *strength* is 0. However, if the structural agent does not find similarity (lexical or semantic) between the super-classes of the compared terms, it is because the terms can be not mapped (terms occurs in different contexts). Then, the *strength* for no mapping is 1. Otherwise, if the structural finds similarity between the super-classes of the compared terms, it is because they can be mapped, but it does not mean that the terms are synonymous, then the *strength* for the mapping is 0. For instance, for terms "Publication/Topic" and "Publication/Proceedings", the structural agent indicates that the terms can be mapped because they have the same super-class, but not with *certainty* because it is no able to indicate that term are similar (mapping with *strength* equals to 0). Otherwise, for the terms "Digital-Camera/Accessories" and "Computer/Accessories", the agent can indicate that the terms can not be mapped because they occur in different contexts (no-mapping with *strength* equal to 1).

3.2. Preferred extension generation

After generating their set of arguments, the agents exchange with each other their arguments. Following a specific protocol, an agent asks (*ask* sign) the others about their arguments. The other agents then, send their arguments to the first agent. An *ack* sign is then sent to requesting agents, in order to indicate that the arguments have been correctly received. Otherwise, an *error* sign is sent.

When all agents have received the set of arguments of each other, they generate their *attacks* set. An *attack* (or counter-argument) will arise when we have arguments for the mapping between the same terms, but with conflicting values of *h*. For instance, an argument $x = (m_1,L,+)$ have as an *attack* an argument $y = (m_2,E,-)$, where m_1 and m_2 refer to the same terms in the ontologies. The argument y also represents an *attack* to the argument x.

As an example, consider the mapping between the terms "Subject" and "Topic", and the lexical and semantic agents. The lexical agent generates an argument $x = (m,L,0)$, where $m = (subject_S,topic_S,-)$; and the semantic agent generates an argument $y = (m,E,1)$, where $m = (subject_S,topic_S,+)$. For both lexical and semantic audiences, the set of arguments is $AR= \{x,y\}$ and the *attacks* $= \{(x,y),(y,x)\}$. However, the relations of *successful attacks* will be defined according to specific audience (see *Definition 2.3.2*), as it is commented below.

When the set of arguments and attacks have been produced, the agents need to define which of them must be accepted. To do this, the agents compute their preferred extension, according to the audiences and *strength*. A set of arguments is *globally subjec-*
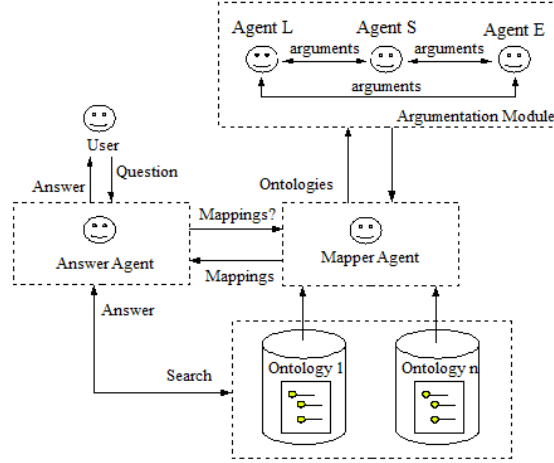
Figure 1. Architecture of a question answering system.

tively acceptable if each element appears in the preferred extension for some agent. A set of arguments is *globally objectively acceptable* if each element appears in the preferred extension for every agent. The arguments which are neither objectively nor subjectively acceptable are considered *indefensible*.

In the example above, considering the lexical(L) and semantic(S) audiences, where $L \succ S$ and $S \succ L$, respectively. For the lexical audience, the argument y successful attacks the argument x, while the argument x does not successful attack the argument y for the semantic audience. Then, the preferred extension of both lexical and semantic agents is composed by the argument y, which can be seen as globally *objectively* acceptable. The mapping between the terms subject$_S$ and topic$_S$ indicated by y is correct.

4. Ontology Mapping in a Question Answering System

A QA system should be able to answer queries in natural language, based on information conveyed by a collection of documents. The answer to a specific question is a set of words and the identification of the document and sentence, which was used as the source of information. In order to answer user queries from heterogeneous data sources described by their own domain specific ontologies, mapping between the ontologies is required.

Figure 1 shows the generic architecture of a web question answering system. We assume that in the context of this question answering system the dynamic nature of the data sources does not make possible to create the mapping a-priory, but mappings need to be created on the fly. Initially, the *answer agent* processes the question sent by the user. In order to answer the question, the *mapper agent* is asked about the mappings between the available data sources. The mappings are obtained from argumentation among specialized agents (lexical, semantic, and structural). The final mappings are then used for the *answer agent* to search the answer in the data sources.
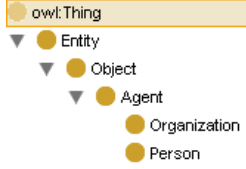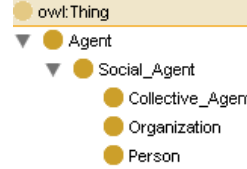
Figure 2. LKIF ontology (partial view).

Figure 3. CLO ontology (partial view).

5. A Walk thought Example

Let us consider that two knowledge bases are described using the LRI-Core (Figure 2) and CLO ontologies (Figure 3), respectively. Three agents need to obtain a consensus about mappings that link corresponding class names in these ontologies. We considered lexical (L), semantic (S), and structural (E) agents in order to verify the behavior of our argumentation model. These agents were implemented in Java 5.0, and the experiments ran on Pentium(R) 4, UCP 3.20GHz, 512MB. Table 1 shows the arguments and counter-arguments (attacks), for the mappings returned by our model.

Table 1. Arguments and attacks.

ID	Argument	At.
1	(Agent,Agent,+,L,1.0)	-
2	(Agent,Agent,+,S,1.0)	-
3	(Agent,Agent,-,E,0.0)	1,2
4	(Agent,Collective-Agent,+,L,0.5)	-
5	(Agent,Collective-Agent,+,S,0.5)	-
6	(Agent,Collective-Agent,-,E,0.0)	4,5
7	(Agent,Social-Agent,+,L,0.5)	-
8	(Agent,Social-Agent,+,S,0.5)	-
9	(Agent,Social-Agent,-,E,0.0)	7,8
10	(Organization,Organization,+,L,1.0)	-
11	(Organization,Organization,+,S,1.0)	-
12	(Organization,Organization,+,E,0.0)	-
13	(Person,Person,+,L,1.0)	-
14	(Person,Person,+,S,1.0)	-
15	(Person,Person,+,E,0.0)	-

As shown in Table 1, the preferred extensions of the agents are composed by the arguments generated by the corresponding audience. The preferred extension of the lexical agents (for all mapped terms) is {1, 4, 7, 10, 13}; the preferred extension of the semantic agent is {1, 4, 7, 10, 13}; and the preferred extension of the structural agent is {1, 4, 7, 10, 13}). We can observe that the arguments 1 and 2; 4 and 5; 7 and 8; 10 and 11; and 13, 14 and 15 refers to the same mapping (the agents agree on the same mapping). When take into account the *"objectively acceptable"* arguments, the arguments 1, 4, 7, 10, and 13 can be considered as consensus. Considering the *strength* of the arguments, the mappings 1,10 and 13 can be observed that the agents have "certainty" in these mappings. For the mappings 4 and 7, the mapped terms could be related through the hyponym relation: "Agent" as super-class of "Social-Agent" and "Collective-Agent".

6. Final Remarks and Future Work

Finding corresponding points between legal ontologies may be of great help in many applications. This paper presented a composite mapping approach based on the argumentation formalism to map legal core ontologies. A review on ontology for legal domain was presented. It shows the diversity of ontologies for this same domain, which illustrates the importance of mapping approaches.

We use an extended argumentation framework (VAF) which associates to each argument a *strength*, representing the confidence that a specific agent has in that argument. We assumed that the confidence degrees is a criteria which is necessary to represent the ontology mapping domain. We have used different agents' output which use distinct mapping algorithms in order to verify the behavior of our model. Partial views of two legal core ontologies, LKIF and CLO were used. We point out that our approach is not restrict to legal domain. The proposed argumentation model seems to be useful for general ontology mapping (see, for example [24][25], where we applied our model for other domains).

In the future, we intend to develop further tests considering also agents using constraint-based mapping approaches (i.e., the similarity between two terms can be based on the equivalence of data types and domains, of key characteristics, or relationship cardinality); use the ontology's application context in our mapping approach (i.e, how the ontology entities are used in some external context, which is especially interesting, for instance, to identify WordNet senses that must be considered to specific terms); and test our approach for less high-level ontologies. Moreover, we plan to extend our model to multilingual ontology mapping. Next, we will use the mapping result as input to an ontology merge process in a question answering system for the law domain.

References

[1] V. K. R. A. *Frame-Based Conceptual Models of Statute Law*. Kluwer Law International, 1995.

[2] J. Bagby and T. Mullen. Legal ontology of sales law application to ecommerce. *Artif. Intell. Law*, 15(2):155–170, 2007.

[3] T. Bench-Capon. Persuasion in practical argument using value-based argumentation frameworks. *Journal of Logic and Computation*, 13:429–448, 2003.

[4] A. Boer and T. Engers. A knowledge engineering approach to comparing legislation. In Springer, editor, *Proceedings of the Fourteenth IFIP International Working Conference*, pages 139–150, 2003.

[5] J. Breuker, A. Elhag, E. Petkov, and R. Winkels. Ontologies for legal information serving and knowledge management. In W. L. Publishers, editor, *Legal Knowledge and Information Systems. Jurix 2002: The Fifteenth Annual Conference*, pages 13–24, 2002.

[6] J. Breuker and R. Hoekstra. Core concepts of law: taking common-sense seriously. In *Proceedings of Formal Ontologies in Information Systems (FOIS-2004)*, 2004.

[7] J. Delgado, I. Gallego, S. Llorente, and R. Garcia. Ipronto: An ontology for digital rights management. In IOPress, editor, *Legal Knowledge and Information System - The Sixteenth Annual Conference: JURIX*, pages 111–120, 2003.

[8] S. Despres and S. Szulman. Construction of a legal ontology from a european community legislative text. In IOPress, editor, *Legal Knowledge and Information System - The Seventeenth Annual Conference: JURIX*, pages 79–88, 2004.

[9] P. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n–person games. *Artificial Intelligence*, 77:321–358, 1995.

[10] A. Gangemi, D. M. Pisanelli, and G. Steve. A formal ontology framework to represent norm dynamics. In *Proceedings of the Second International Workshop on Legal Ontologies*, 2001.

[11] A. Gangemi, D. M. Pisanelli, and G. Steve. A formal ontology framework to represent norm dynamics. In *Congreso Internacional de Culturas y Sistemas Jurídicos Comparados*, 2005.

[12] T. R. Gruber. For the automated mark-up of Italian legislative texts in XML. In B. Capon, editor, *Legal Knowledge and Information System - The Fifteenth Annual Conference: JURIX*, pages 21–30. IOS Press, Amsterdam, Netherlands, London, UK, 1993.

[13] F. Hakimpour and A. Geppert. Resolving semantic heterogeneity in schema integration: an ontology approach. In *Proceedings of the International Conference on Formal Ontology in Informational Systems*, 2001.

[14] R. Hoekstra, J. Breuker, M. D. Bello, and A. Boer. The LKIF Core ontology of basic legal concepts. In P. Casanovas, M. A. Biasiotti, E. Francesconi, and M. T. Sagri, editors, *Proceedings of the Workshop on Legal Ontologies and Artificial Intelligence Techniques (LOAIT 2007)*, June 2007.

[15] R. Laarschot, W. Steenbergen, H. Stuckenschmidt, A. Looder, and F. Harmelen. The legal concepts and the layman's terms. In IOPress, editor, *Legal Knowledge and Information System - The 18th Annual Conference: JURIX*, 2005.

[16] L. Laera, V. Tamma, J. Euzenat, T. Bench-Capon, and T. R. Payne. Reaching agreement over ontology alignments. In *Proceedings of 5th International Semantic Web Conference (ISWC 2006)*, 2006.

[17] I. Levenshtein. Binary codes capable of correcting deletions, insertions an reversals. In *Cybernetics and Control Theory*, 1966.

[18] A. Maedche and S. Staab. Measuring similarity between ontologies. In *Proceedings of the European Conference on Knowledge Acquisition and Management*, pages 251–263, 2002.

[19] L. T. McCarty. A language for legal discourse i. basic features. In *ICAIL '89: Proceedings of the 2nd international conference on Artificial intelligence and law*, pages 180–189, New York, NY, USA, 1989. ACM Press.

[20] J. Saias and P. Quaresma. A methodology to create legal ontologies in a logic programming information retrieval system. In *Law and the Semantic Web*, pages 185–200, 2003.

[21] J. Shaheed, A. Yip, and J. Cunningham. A top-level language-biased legal ontology. In W. L. Publishers, editor, *Legal Ontologies and Artificial Intelligence Techniques*, pages 13–24, 2005.

[22] R. K. Stamper. The role of semantics in legal expert systems and legal reasoning. *Ration Juris*, 4(2), 1991.

[23] G. Stoilos, G. Stamou, and S. Kollias. A string metric for ontology alignment. In *ISWC*, pages 624–637. 4th International Semantic Web Conference (ISWC 2005), 2005.

[24] C. Trojahn, P. Quaresma, and R. Vieira. A cooperative approach for composite ontology mapping. *LNCS Journal of Data Semantic (to appear)*, 2007.

[25] C. Trojahn, P. Quaresma, and R. Vieira. An extended value-based argumentation framework for ontology mapping with confidence degrees. In *Workshop of Argumentation, International Conference on Autonomous Agents and Multiagent Systems*, 2007.

[26] A. Valente. *Legal Knowledge Engineering, Modelling Approach (Frontiers in Artificial Intelligence and Applications, 30)*. IOS Press, January 1995.

[27] P. R. S. Visser. *Knowledge Specification for Multiple Tasks*. Kluwer Law International, 1995.