2016 IEEE Tenth International Conference on Semantic Computing

# Towards A Meta-reasoning Framework for Reasoning about Vagueness in OWL Ontologies

Nophadol Jekjantuk
*Department of Computing Science*
*University of Aberdeen*
*Aberdeen, United Kingdom*
*Email: n.jekjantuk@abdn.ac.uk*

Jeff Z. Pan
*Department of Computing Science*
*University of Aberdeen*
*Aberdeen, United Kingdom*
*Email: jeff.z.pan@abdn.ac.uk*

Panos Alexopoulos
*Expert System Iberia*
*Madrid, Spain*
*Email: palexopoulos@expertsystem.com*

***Abstract*—When developing ontologies, knowledge engineers and domain experts often use predicates that are vague, i.e., predicates that lack clear applicability conditions and boundaries such as *High*, *Expert* or *Bad*. In previous works, we have shown how such predicates within ontologies can hamper the latter's shareability and meaning explicitness and we have proposed Vagueness Ontology (VO), an OWL metaontology for representing vagueness-aware ontologies, i.e., ontologies whose (vague) elements are annotated by explicit descriptions of the nature and characteristics of their vagueness. A limitation of VO is that it does not model the way vagueness and its characteristics propagate when defining more complex OWL axioms (such as conjunctive classes), neither does it enforce any kind of vagueness-related consistency. For that, in this paper, we expand VO by means of formal inference rules and constraints that model the way vagueness descriptions of complex ontology elements can be automatically derived. More importantly, we enable the efficient execution of these rules by means of a novel meta-reasoning framework.**

***Keywords*-Vague Concept; Vagueness Ontology; Metamodeling Reasoning; Ontological Metamodeling;**

## I. Introduction

Vagueness is a common human language phenomenon, typically manifested by terms and concepts like High, Expert, Bad, Near etc., and related to our inability to precisely determine the extensions of such concepts in certain domains and contexts. That is because vague concepts have typically blurred boundaries which do not allow for a sharp distinction between the entities that fall within their extension and those that do not [11] [17]. For example, some people are borderline tall: not clearly "tall" and not clearly "not tall".

When building ontologies, engineers and domain experts often define ontology entities (classes, relations, etc.) that contain vague predicates. The problem with this practice is that the existence of such predicates, as we have illustrated in previous work [2], can influence in a negative way the comprehension of ontologies by other parties and limit their value as a reusable source of knowledge. The reason is the subjective interpretation of such definitions that may cause **disagreements** among the people who develop, maintain or use an ontology.

In the same work, we have put forward the notion of **vagueness-aware ontologies** as a way to reduce the number and intensity of such disagreements. Informally, a vagueness-aware ontology is defined as *"an ontology whose vague entities are accompanied by comprehensive metainformation that describes the nature and characteristics of their vagueness"*. A simple example of such metainformation is whether an ontology entity (e.g., a class) is vague or not; this is important as many ontology users may not immediately realize this. A more sophisticated example is the particular type of the entity's vagueness or the applicability context of its definition. In all cases, the rationale is that having such metainformation, explicitly represented and published along with (vague) ontologies, can improve the latter's comprehensibility and shareability, by narrowing the possible interpretations that its vague entities may assume by human and software agents.

To enable the formal representation of vagueness-aware ontologies we have developed the **Vagueness Ontology (VO)**[1] [1], a metaontology that defines the necessary concepts, relations, and attributes for creating explicit descriptions of vague ontology entities. VO is meant to be used by both producers and consumers of ontologies; the former will utilize it to **annotate** the vague part of their produced ontologies with relevant vagueness metainformation while the latter will **query** this metainformation and use it to make a better use of the vague ontologies.

A limitation of the Vagueness Ontology is that it does not model the way vagueness and its characteristics propagate when defining more complex OWL axioms neither does it enforce any kind of vague-related consistency. For example, if we define an OWL class as the conjunction of a set of vague and non-vague classes, then the reasoner should infer that this class is also vague. Similarly, if we define a vague object property to be transitive, the reasoner should indicate a problem as vague properties cannot be transitive [19]. To facilitate such capabilities, in this paper, we a) extend the Vagueness Ontology by means of formal constraints and

[1] http://www.essepuntato.it/2013/10/vagueness

978-1-5090-0662-5/16 $31.00 © 2016 IEEE
DOI 10.1109/ICSC.2016.47

inference rules that model the way vagueness annotations of complex ontology elements can be automatically derived from simpler elements and b) implement and evaluate a novel meta-reasoning framework that enables the efficient execution of these rules within OWL. In practice, these rules allow the automatic derivation of vagueness descriptions for indirectly defined ontology elements and they enforce a kind of vagueness-related consistency within the ontology.

The structure of the rest of the paper is as follows. The next section describes the Vagueness Ontology and introduces some meta-reasoning rules that apply to it. Section III, in turn, describes our approach for facilitating the execution of these rules on VO-annotated ontologies in an efficient way while section IV describes an evaluation of this approach. Section V presents related work and finally, section VI concludes.

## II. Defining Meta-reasoning Rules in the Vagueness Ontology

### A. The Vagueness Ontology

The Vagueness Ontology enables the annotation of an ontological entity (class, relation or datatype) with a description of the nature and characteristic of its vagueness. A class is vague if, in the given domain, it admits borderline cases, namely if there are (or could be) individuals for which it is indeterminate whether they instantiate the class (e.g., *"TallPerson"*, *"ExperiencedResearcher"*, etc.). Similarly, an object property (relation) is vague if there are (or could be) pairs of individuals for which it is indeterminate whether hey stand in the relation (e.g., *"hasGenre"*, *"hasIdeology"*, etc.). The same applies for datatype properties and pairs of individuals and literal values. Finally, a vague datatype consists of a set of vague terms (e.g., *"RestaurantPriceRange"* with the terms *"cheap"*, *"moderate"* and *"expensive"*).

A vagueness description explicitly states whether the entity is vague or not. For example, the class *"StrategicClient"* defined as *"A client that has a high value for the company"* is vague while *"AmericanCompany"* as *"A company that has legal status in the Unites States"* is not. Moreover, it can often be the case that a seemingly vague entity can have a non-vague definition (e.g. *"TallPerson"* when defined as *"A person whose height is at least 180cm"*). Then this entity is not vague in the given ontology and that is something that needs to be explicitly stated.

Also, vagueness can be **quantitative** or **qualitative** [11]. A predicate has quantitative vagueness if the existence of borderline cases stems from the lack of precise boundaries for the predicate along one or more **dimensions** (e.g. "*bald*" lacks sharp boundaries along the dimension of hair), and qualitative if there is a variety of conditions pertaining to the predicate, but it is not possible to make any crisp identification of those combinations which are sufficient for application (e.g., "*religion*", "*strategic*", etc.).

Knowing the type of vagueness is important as entities with an intended (but not explicitly stated) quantitative vagueness can be considered by others as having qualitative one and vice versa. Also, when the entity has quantitative vagueness it is important to state explicitly its intended dimensions (e.g. the amount of R&D budget for the term "*strategic*". VO makes explicit the type of the entity's vagueness and the dimensions of the term's quantitative vagueness. Moreover, VO explicitly represents the **creator** of a vagueness annotation of a certain entity as well as the **applicability context** for which the entity is defined.

Context-dependent can be i) the description of vagueness of an entity (i.e. the same entity can be vague in one context and non-vague in another) and ii) the dimensions related to a description of vagueness having quantitative type (i.e. the same entity can be vague in dimension A in one context and in dimension B in another).

As an example assume that Silvio Peroni thinks that the class TallPerson is vague since there is no way to define a crisp height threshold that may separate tall from non-tall people. Using the Vagueness Ontology, this information could represented as follows:

```
:TallPerson a owl:Class.

:silvio-peroni a :prov:Agent.

:silvio-vag-anno-for-tall-person a vag:
    VaguenessAnnotation;
        oa:hasBody :silvio-vag-desc-for-tall-
            person;
        oa:hasTarget :TallPerson;
        prov:wasAttributedTo :silvio-peroni.

:silvio-vag-desc-for-tall-person a vag:
    DescriptionOfVagueness;
        vag:hasJustification :silvio-jus-for-vag-
            of-tall-person.

:silvio-jus-for-vag-of-tall-person a vag:
    Justification;
        vag:hasNaturalLanguageText  "Tall Person
            is vague because it's not possible to
            define the height threshold between
            tall and non-tall.".
```

Such representations allow us to ask questions related to the vagueness of a given ontology in the form of SPARQL queries. For example, to retrieve all vague entities of a VO-annotated ontology, the following query could be used:

```
SELECT DISTINCT ?entity WHERE {
 ?ann a vag:VaguenessAnnotation ;
       oa:hasTarget ?entity ;
       oa:hasBody ?desc .
 ?desc a vag:DescriptionOfVagueness }
```

### B. Rules and Constraints about Vagueness Descriptions

The annotation of vague ontology elements with vagueness descriptions is typically done in a manual fashion by the ontology engineer. When these elements participate in the definition of more complex elements, such as conjunctive classes, then their descriptions can be used to automatically infer vagueness descriptions for these complex elements. Moreover, it may be the case that an element's vagueness description that an engineer or domain expert creates is inconsistent with the other characteristics of the element; in such a case this inconsistent should be detected. The identification of vagueness-related inference rules and constraints with respect to the Vagueness Ontology is a work in progress; so far we have identified the following:

- VR 1: The subclass of a quantitatively vague class is also vague unless this subclass provides concrete thresholds for the vagueness dimensions of the class. The reason for this is that the vagueness dimensions of the class are inherited by the subclass, meaning that the latter can become non-vague only if it provides specific values for these dimensions. For example the class PersonOver180cm is a subclass of the vague class TallPerson but itself is not vague because it makes concrete the height threshold.
- VR 2: The subclass of a qualitatively vague class is also vague unless this subclass consists of instances that satisfy a subset of the class's vague applicability conditions. Again the rationale here is that the subclass inherits the lack of concrete applicability criteria of the parent class and can only be non-vague if it explicitly provides these criteria. For example, assuming that a criterion for a company to be a competitor for a Spanish company is to have offices in Spain, the class CompanyWithOfficesInSpain is a subclass of the vague class Competitor without itself being vague.
- VR 3: The conjunction of a set of classes is vague if at least one of these classes is vague. For example, if TallGreekPerson $\equiv$ TallPerson $\sqcap$ GreekPerson and the classes TallPerson and GreekPerson have been identified as vague and non-vague respectively, then TallGreekPerson is also vague.
- VR 4: The conjunction of a set of classes is quantitatively vague if all these classes are quantitatively vague. The reason for this is that if one of the constituent vague classes has qualitative vagueness, then the criterion for quantitative vagueness (i.e., being vague only in particular dimensions) does not apply for the conjunctive class.
- VR 5: The conjunction of a set of classes is qualitatively vague if at least one of these classes is qualitatively vague. Again the reason for this is that the qualitative vagueness criterion dominates the quantitative one.
- VR 6: The conjunction of a set of quantitatively vague classes is vague in the super set of all the constituent classes' dimensions. For example, if TallFatPerson $\equiv$ TallPerson $\sqcap$ FatPerson and the classes TallPerson and FatPerson are quantitatively vague in the dimensions of Height and Weight respectively, then TallFatPerson is quantitatively vague in the both of these dimensions.
- VR 7: A vague property cannot be transitive. The reason for this is that vagueness destroys transitivity. For example, if $x$ is approximately equal to $y$ and $y$ approximately equal to $z$, then it does not necessarily follow that $x$ is approximately equal to $z$ [19].
- VR 8: The inverse of a vague property has the same vagueness characteristics as the original property.

In the next section, we describe how we facilitate the execution of the above rules in an efficient way by means of a reasoning algorithm that builds on top of an existing DL reasoner. Please note that the first two rules are not currently possible to implement because of the expressivity of the OWL 2 DL; therefore, we implement here only rules VR 3 to VR 8.

## III. Executing Meta-reasoning Rules in the Vagueness Ontology

### A. OWL 2 (Recap)

An OWL 2 DL vocabulary $\mathcal{V}_{\mathcal{O}}$ = ($\mathcal{V}_{cls}$, $\mathcal{V}_{op}$, $\mathcal{V}_{dp}$, $\mathcal{V}_{ind}$, $\mathcal{V}_{dt}$, $\mathcal{V}_{lt}$, $\mathcal{V}_{fa}$) is a 7-tuple over a datatype map $D$ where $\mathcal{V}_{cls}$ is the set of IRIs denoting class names, $\mathcal{V}_{op}$ is the set of IRIs denoting object properties, $\mathcal{V}_{dp}$ is the set of IRIs denoting datatype properties, $\mathcal{V}_{ind}$ is the set of IRIs denoting individuals, $\mathcal{V}_{dt}$ is the set of IRIs denoting all datatypes of $D$, the datatype rdfs:Literal, and possibly other datatypes, is the set of IRIs denoting datatype names, $\mathcal{V}_{lt}$ is the set of well-formed RDF literals and $\mathcal{V}_{fa}$ is the set of pairs ($F$, $lt$) for each containing facet $F$, and literal $lt$.

The abstract syntax for an OWL 2 class definition is:

$$C \leftarrow \top \mid \bot \mid CN \mid \neg C \mid C \sqcap D \mid C \sqcup D \mid \{o\} \mid \\ \exists R.C \mid \forall R.C \mid \exists R.Self \mid \leq nR.C \mid \geq nR.C$$

where $C \in \mathcal{V}_{cls}$, $a \in \mathcal{V}_{ind}$, $R, T \in \mathcal{V}_{op}$ and it are simple roles, and $n$ is a non-negative integer.

The semantics of OWL 2 ontologies is given by means of interpretations. An interpretations $\mathcal{I}$ consists of a set $\Delta^{\mathcal{I}}$ called domain together with a function $\cdot^{\mathcal{I}}$ mapping individual names to elements of $\Delta^{\mathcal{I}}$, class names to subsets of $\Delta^{\mathcal{I}}$, and role names to subsets of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. The interpretation function is extended to complex class expressions in the usual way as explained in detail in [10].

### B. OWL FA

OWL FA [15] enables metamodeling. It is an extension of OWL DL, which refers to the description logic $\mathcal{SHOIN}(\mathcal{D})$. Ontologies in OWL FA are represented in a layered architecture.

OWL FA specifies a stratum number in class constructors and axioms to indicate the strata they belong to. Let $i \geq 0$ be an integer. OWL FA consists of an alphabet of distinct class names $\mathbf{V}_{\mathrm{C_i}}$ (for stratum i), datatype names $\mathbf{V}_{\mathrm{D}}$, abstract property names $\mathbf{V}_{\mathrm{AP_i}}$ (for stratum i), datatype property names $\mathbf{V}_{\mathrm{DP}}$ and individual (object) names ($\mathbf{I}$); together with a set of constructors (with subscriptions) to construct class and property descriptions (also called *OWL FA-classes* and *OWL FA-properties*, respectively).

Let $\mathsf{CN} \in \mathbf{V}_{\mathrm{C_i}}$ be an atomic class name in layer $i$ ($i \geq 0$), $R$ an OWL FA-property in layer $i$, $o \in \mathbf{I}$ an individual, $T \in \mathbf{V}_{\mathrm{DP}}$ a datatype property name, and $C, D$ OWL FA-classes in layer $i$. Valid OWL FA-classes are defined by the abstract syntax:

$$\begin{aligned} C \leftarrow\ & \top \mid \bot \mid CN \mid \neg_i C \mid C \sqcap_i D \mid C \sqcup_i D \mid \{o\} \mid \\ & \exists_i R.C \mid \forall_i R.C \mid \leq_i nR \mid \geq_i nR \mid (if\ i = 1) \\ & \exists_1 T.d \mid \forall_1 T.d \mid \leq_1 nT \mid \geq_1 nT \end{aligned}$$

The semantics of OWL FA is a model theoretic semantics, which is defined in terms of interpretations. In other words, the semantics of two layers which can be considered as TBox and ABox are same as in OWL DL. The idea of OWL FA is that the interpretation depends on the layer but is still an OWL DL interpretation. Given an OWL FA alphabet $\mathbf{V}$, a set of built-in datatype names $\mathbf{B} \subseteq \mathbf{V}_{\mathrm{D}}$ and an integer k $\geq 1$, an *OWL FA interpretation* is a tuple of the form pair $\mathcal{J} = (\Delta^{\mathcal{J}}, \cdot^{\mathcal{J}})$, where $\Delta^{\mathcal{J}}$ is the domain (a non-empty set) and $\cdot^{\mathcal{J}}$ is the interpretation. In the rest of the paper, we assume that $i$ is an integer such that $1 \leq i \leq k$. The interpretation function can be extend ObjectProperty assertionsed to give semantics to OWL FA-properties and OWL FA-classes. Let $RN \in \mathbf{V}_{\mathrm{AP_i}}$ be an abstract property name in layer $i$ and $R$ be an abstract property in layer $i$. Valid OWL FA abstract properties are defined by the abstract syntax: $R ::= RN \mid R^{-}$, where for some $x, y \in \Delta_{A_{\mathrm{i-1}}^{\mathcal{J}}}$, $\langle x, y\rangle \in R^{\mathcal{I}}$ iff $\langle y, x\rangle \in R^{-\mathcal{J}}$. Valid OWL FA datatype properties are datatype property names. The interpretation function is explained in detail in [15].

### C. Classed-based Metamodeling Approach

Classed-based Metamodeling [9] is ontology-inherent metamodeling for classes in OWL based on an axiomatization of class reification. We can obtain the a metamodeling-enabled version $\mathcal{O}^{meta}$ from a given ontology $\mathcal{O}$ with function $\mathsf{bound}(\cdot)$,$\mathsf{SepDom}(\cdot)$,$\mathsf{Typing}$ $(\cdot)$ and $\mathsf{MatSubClass}(\cdot)$.

Let $\mathcal{O}$ be a domain ontology with vocabulary $\mathcal{V}(\mathsf{O}_C, \mathsf{O}_R, \mathsf{O}_I)$ The vocabulary of the metamodeling-enabled version is

$$\begin{aligned} \mathsf{O}_C^{meta} &:= \mathsf{O}_C \cup \{\mathsf{Inst}, \mathsf{Class}\} \\ \mathsf{O}_R^{meta} &:= \mathsf{O}_R \cup \{\mathsf{type}, \mathsf{subClassof}, R_{\mathsf{Inst}}\} \\ \mathsf{O}_I^{meta} &:= \mathsf{O}_I \cup \{o_{\mathsf{C}} \mid \mathsf{C} \in \mathsf{O}_C\} \end{aligned}$$

where all newly introduced names are fresh where they are not part of $\mathcal{V}_{\mathcal{O}}$.

The function $\mathsf{bound}(\cdot)$ returns its input after rewriting it as follows: first, every occurrence of $X$ having one of the forms $\top$, $\neg\mathsf{C}$, $\forall R.\mathsf{C}$, $\leqslant nR.\mathsf{C}$, $\exists U.\mathsf{Self}$ is substituted by $\mathsf{Inst} \sqcap X$, where we can express complex classes C. Next, the universal role is localized by substituting every $\forall U.\mathsf{C}$ by $\forall U.(\mathsf{Inst} \sqcup \mathsf{C})$ and every U occurring on the left hand side of a role chain axiom by $R_{\mathsf{Inst}} \circ U \circ R_{\mathsf{Inst}}$ where $R_{\mathsf{Inst}}$ is axiomatized via $\exists R_{\mathsf{Inst}}.\mathsf{Self} \equiv \mathsf{Inst}$. The functions SepDom, Typing, and MatSubClass return a set of axioms as specified in [9]. The metamodeling-enabled version $\mathcal{O}^{meta}$ of $\mathcal{O}$ is $\mathsf{bound}(\mathcal{O}) \cup \mathsf{SepDom}(\mathcal{O}) \cup \mathsf{Typing}(\mathcal{O}) \cup \mathsf{MatSubClass}(\mathcal{O})$.

### D. Extended Fixed-Layer Architecture Reasoning

As we discussed in section II-B, vagueness rules 3 - 6 are only apply to the axiom of form $\mathsf{A} \equiv \sqcap \mathsf{B_n}$ where $n \geq 1$. This form of axiom can be transformed into $\mathsf{A} \sqsubseteq \mathsf{B}_1, \ldots, \mathsf{A} \sqsubseteq \mathsf{B_n}$. Thus, the subsumption relation will be useful to help us to identify the hidden vague entity.

According to the layered architecture, the knowledge base $\mathcal{O}$ in OWL FA is divided into a sequence of knowledge bases $\mathcal{O} = \mathcal{O}_1, \ldots \mathcal{O}_{\mathrm{k}}$, whereas $k$ is the number of layers. We have extended Fixed-Layer Architecture semantics with role name subClassOf and subPropertyOf. Subclass relationships between a class C and a class D in the given ontology $\mathcal{O}_i$ are materialized as role instances: the individual that represents the class C, say $\mathsf{O}_C$, and the individual that represents D, say $\mathsf{O}_D$, are interconnected by the newly introduced metarole subClassOf in $\mathcal{O}_{i+1}$ . Similar to SubProperty relationships between a property P and a property Q are materialized as role instances of metarole subPropertyOf.

**Definition 1.** *Let $\mathcal{O}_i$ be an ontology in a set of ontologies of OWL FA and i > 1. $\mathcal{O}_i$ consist of vocabulary $\mathcal{V}(\mathsf{O}_C, \mathsf{O}_R, \mathsf{O}_I)$. The vocabulary of the extension version of $\mathcal{O}_i$ is*

$$\mathsf{O}_R := \mathsf{O}_R \cup \{\mathsf{subClassOf}, \mathsf{subPropertyOf}\}$$

*where* subClassOf *and* subPropertyOf *newly introduced role names and they are not part of $\mathcal{V}_{\mathcal{O}}$.*

An interpretation $\mathcal{J}$ satisfies $\mathsf{subClassOf}^{\mathcal{J}} = \{< \delta_C, \delta_D > \mid C^{\mathcal{I}} \subseteq D^{\mathcal{I}}\}$ and $\mathsf{subPropertyOf}^{\mathcal{J}} = \{< \delta_R, \delta_S > \mid R^{\mathcal{I}} \subseteq S^{\mathcal{I}}\}$.

**Theorem 1.** *Let $\mathcal{O}_i$ be an OWL ontology on layer i where i >1. Then, the extend of $\mathcal{O}_i$ contains class/role subsumption constrains from $\mathcal{O}_{i-1}$ as role instances according to their semantics rules.*

*Proof:* (sketch). The theorem is proven by an obvious interpretation to satisfies meta role $\mathsf{subClassOf}(\delta_C, \delta_D)$ iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ and $\mathsf{subPropertyOf}(\delta_R, \delta_S)$ iff $R^{\mathcal{I}} \subseteq S^{\mathcal{I}}$. ■

Reasoning problems of the extended FA semantics can be reduced to DL reasoning problems by propagating equality and subsumption constraints between $\mathcal{O}_i$ and $\mathcal{O}_{i+1}$. However, reasoning tasks could be expensive because we have to propagate up- and downward until there is no new axiom generated. Since, we possess only a domain ontology and the vagueness metaontology, we can use the class-based approach to encode a domain ontology into the Vagueness

Ontology. Thus, the reasoning would be simpler because it does not require the propagation between layers. In the next section, we are going to discuss how to extend a class-based approach to integrate a domain ontology with the Vagueness Ontology base on the extended FA semantics.

### E. Integrated Domain Ontology

In this section, we show how to extend a class-based approach [9] to integrate a domain ontology with the Vagueness Ontology. The Class is a newly introduced class which contains class individuals that represent classes from domain ontology. The Role contains property individuals that represent properties from a domain ontology excluding transitive role and RoleTran contains property individuals that represent all transitive roles.

**Definition 2.** *Let $\mathcal{O}$ be a domain ontology with vocabulary $\mathcal{V}(\mathsf{O}_C, \mathsf{O}_R, \mathsf{O}_I)$ The vocabulary of a metamodeling-enabled version $\mathcal{O}^{meta}$ is*

$$\mathsf{O}_C^{meta} = \mathsf{O}_C \cup \{\mathsf{Class}, \mathsf{Role}, \mathsf{RoleTran}, \mathsf{Inst}\}$$
$$\mathsf{O}_R^{meta} = \mathsf{O}_R \cup \{\mathsf{subClassof}, \mathsf{subPropertyOf}, \mathsf{R}_{Inst}\}$$
$$\mathsf{O}_I^{meta} = \mathsf{O}_I \cup \{o_\mathsf{C} \mid \mathsf{C} \in \mathsf{O}_C, o_\mathsf{R} \mid \mathsf{R} \in \mathsf{O}_R\}$$

*where all newly introduced names are fresh where they are not part of $\mathcal{V}_\mathcal{O}$.*

We use function $bound(\cdot)$ and $SepDom(\cdot)$ which defined in [9] to compress the domain ontology in order to integrate it with the Vagueness Ontology. We have extended the function $SepDom(\cdot)$ with following rules:

- Class $\sqsubseteq$ $\neg$Role
- RoleTran $\sqsubseteq$ Role
- $\exists\mathsf{subPropertyOf}.\top \sqsubseteq \mathsf{Role}$
- $\top \sqsubseteq \forall\mathsf{subPropertyOf}.\mathsf{Role}$
- Role($o_\mathsf{R}$) for all $\mathsf{R} \in \mathsf{O}_R$ excluding transitive role
- RoleTran($o_\mathsf{R}$) for all transitive role $\in \mathsf{O}_R$

We have dropped the functions $Typing(\cdot)$ and $MatSubClass(\cdot)$ and we substitute these functions with the propagation rules as following:

- $C \sqsubseteq D$ and $o_C$, $o_D \in$ Class then add subClassOf($o_C$, $o_D$) to $\Sigma$.
- $C \equiv D$ and $o_C$, $o_D \in$ Class then add $o_C = o_D$ to $\Sigma$.
- $R \sqsubseteq S$ and $o_R$, $o_S \in$ Role then add subPropertyOf($o_R$, $o_S$) to $\Sigma$.
- $R \equiv S$ and $o_R$, $o_S \in$ Role then add $o_R = o_S$ to $\Sigma$.
- $o_C = o_D$ and $o_C$, $o_D \in$ Class then add $C \equiv D$ to $\Sigma$.
- subClassOf($o_C$, $o_D$) and $o_C$, $o_D \in$ Class then add $C \sqsubseteq D$ to $\Sigma$.
- $o_R = o_S$ and $o_R$, $o_S \in$ Role then add $R \equiv S$ to $\Sigma$.
- subPropertyOf($o_R$, $o_S$) and $o_R$, $o_S \in$ Role then add $R \sqsubseteq S$ to $\Sigma$.

**Theorem 2.** *Let $\mathcal{O}$ be an OWL ontology and $\mathcal{O}_{meta}$ be its metamodeling-enabled version as specified in Definition 2. Then the following properties hold:*

- *For any OWL axiom $\alpha$ containing only names from $\mathsf{O}_C$, $\mathsf{O}_R$, and $\mathsf{O}_I$ we have $\mathcal{O} \models \alpha$ iff $\mathcal{O}_{meta} \models bound(\alpha)$.*
- *For any two named classes $C, D \in \mathsf{O}_C$, we have that $\mathcal{O} \models C \sqsubseteq D$ iff $\mathcal{O}_{meta} \models \mathsf{subClassOf}(o_C, o_D)$ .*
- *For any two named classes $C, D \in \mathsf{O}_C$, we have that $\mathcal{O} \models C \equiv D$ iff $\mathcal{O}_{meta} \models o_C = o_D$.*
- *For any two named properties $R, S \in \mathsf{O}_R$, we have that $\mathcal{O} \models R \sqsubseteq S$ iff $\mathcal{O}_{meta} \models \mathsf{subPropertyOf}(o_R, o_S)$.*
- *For any two named properties $R, S \in \mathsf{O}_R$, we have that $\mathcal{O} \models R \equiv S$ iff $\mathcal{O}_{meta} \models o_R = o_S$.*

*Proof:* (sketch). For the first claim, given a model $\mathcal{I}$ of $\mathcal{O}$, we construct a model meta($\mathcal{I}$) = $\mathcal{J}$ of $\mathcal{O}_{meta}$ as follows:

- $\Delta^\mathcal{J} = \Delta^I \cup \{o_\mathsf{C} \mid \mathsf{C} \in \mathsf{O}_C\} \cup \{o_\mathsf{R} \mid \mathsf{R} \in \mathsf{O}_R\}$, $\mathsf{Inst}^\mathcal{J} = \Delta^I$
- $\varsigma^\mathcal{J} = \varsigma^\mathcal{I}$ for all $\varsigma \in \mathsf{O}_C \cup \mathsf{O}_R \cup \mathsf{O}_I$
- $\mathsf{Class}^\mathcal{J} = \{o_\mathsf{C} \mid \mathsf{C} \in \mathsf{O}_C\}$, $\mathsf{Role}^\mathcal{J} = \{o_\mathsf{R} \mid \mathsf{R} \in \mathsf{O}_R$ and R is not transitive$\}$
- $\mathsf{RoleTran}^\mathcal{J} = \{o_\mathsf{RoleTran} \mid$ R is transitive$\}$
- $\mathsf{subClassOf}^\mathcal{J} = \{< \delta_C, \delta_D > \mid C^\mathcal{I} \subseteq D^\mathcal{I}\}$
- $\mathsf{subPropertyOf}^\mathcal{J} = \{< \delta_R, \delta_S > \mid R^\mathcal{I} \subseteq S^\mathcal{I}\}$

we can construct $\mathcal{J}$ satisfies for all axioms from $\mathsf{SepDom}(O)$ and propagation rules. For every class $C$ containing only names from $\mathsf{Voc}(O)$, that $\mathsf{bound}(C)^\mathcal{J} = C^\mathcal{I}$ (claim †).

For the second claim, we have that from $\mathcal{O} \models C \sqsubseteq D$ follows $\mathcal{O}_{meta} \models \mathsf{bound}(C) \sqsubseteq \mathsf{bound}(D)$. Therefore, $\mathcal{O}_{meta} \models C \sqsubseteq D$. Considering a model $\mathcal{J}$ of $\mathcal{O}_{meta}$ and the propagation rules. we obtain $\mathsf{subClassOf}(\delta_C, \delta_D)$ iff $C^\mathcal{I} \subseteq D^\mathcal{I}$ the argument holds in both ways. For the rest of the claim, it can be proofed with similar manner. ■

We now present the algorithm `Compute`, that will propagate equality and subsumption constrains between $\mathcal{O}_{meta}$ and $\mathcal{O}_{VO}$. The algorithm takes $\mathcal{O}_{meta}$ and $\mathcal{O}_{VO}$ as input and returns the explicit knowledge base. The Algorithm `Compute` is shown in Algorithm 1.

### F. Vagueness Ontology Extension

We define OWL based constraints as OWL class definitions instead of a SPARQL query that presented in [1]. VagueEntity is an OWL class definition contains individuals that represent classes/properties that has been identified as vague entity. NonVageEntity, VagueClass, VagueRole, NonVagueClass, NonVagueRole, VaguenessQualitative, VaguenessQuantitative, VaguenessDimension, VaguenessDimensionInContext are class definition that hold individuals which classify into non vague entity, vague class, vague property, non vague class, non vague property, qualitative vagueness, quantitative vagueness, vagueness dimension and vagueness dimension in context respectively.

**Algorithm 1** Compute

**Input:** $\{\mathcal{O}_{meta}, \mathcal{O}_{VO}\}$
**Output:** $\Sigma_{MetOn}$

**procedure** COMPUTE
  Create $\Sigma_{MetOn}$ from $\mathcal{O}_{VO}$ and $\mathcal{O}_{meta}$
  **repeat**
    $C \sqsubseteq D$ and $o_C$, $o_D$ $\in$ Class then add subClassOf($o_C$, $o_D$) to $\Sigma_{MetOn}$
    $C \equiv D$ and $o_C$, co$_D$ $\in$ Class then add $o_C$ = $o_D$ to $\Sigma_{MetOn}$
    $R \sqsubseteq S$ and $o_R$, $o_S$ $\in$ Role then add subPropertyOf($o_R$, $o_S$) to $\Sigma_{MetOn}$
    $R \equiv S$ and $o_R$, $o_S \in$ Role then add $o_R$ = $o_S$ to $\Sigma_{MetOn}$
    $o_C$ = $o_D$ and $o_C$, $o_D \in$ Class then add $C \equiv D$ to $\Sigma_{MetOn}$
    subClassOf($o_C$, $o_D$) and $o_C$, $o_D$ $\in$ Class then add $C \sqsubseteq D$ to $\Sigma_{MetOn}$
    $o_R$ = $o_S$ and $o_R$, $o_S \in$ Role then add $R \equiv S$ to $\Sigma_{MetOn}$
    subPropertyOf($o_R$, $o_S$) and $o_R$, $o_S \in$ Role then add $R \sqsubseteq S$ to $\Sigma_{MetOn}$
  **until** there is no new axiom generated
  **return** $\Sigma_{MetOn}$

- VagueEntity is a class that contains all vague entities.

```
SELECT DISTINCT ?entity WHERE {
 ?ann a vag:VaguenessAnnotation ;
       oa:hasTarget ?entity ;
       oa:hasBody ?desc .
 ?desc a vag:DescriptionOfVagueness }
```

  $\mathsf{VagueEntity} \equiv \exists \mathsf{hasTarget}^{-}.(\mathsf{VaguenessAnnotation} \sqcap (\exists \mathsf{hasBody.DescriptionOfVagueness}))$
- NonVagueEntity is a class that contains all non vague entities.
  $\mathsf{NonVagueEntity} \equiv \exists \mathsf{hasTarget}^{-}.(\mathsf{VaguenessAnno-tation} \sqcap (\exists .\mathsf{hasBody.DescriptionOfNonVagueness}))$
- VagueClass and VagueRole: vague entities of a certain type.
  $\mathsf{VagueClass} \equiv (\mathsf{VagueEntity} \sqcap \mathsf{Class}) \sqcup (\exists \mathsf{subClassOf.VagueClass})$
  $\mathsf{VagueRole} \equiv (\mathsf{VagueEntity} \sqcap \mathsf{Role}) \sqcup (\exists \mathsf{subPropertyOf.VagueRole})$
- NonVagueClass and NonVagueRole: non vague entities of a certain type.
  $\mathsf{NonVagueClass} \equiv \mathsf{NonVagueEntity} \sqcap \mathsf{Class}$
  $\mathsf{NonVagueRole} \equiv \mathsf{NonVagueEntity} \sqcap \mathsf{Role}$
- VaguenessQualitative and VaguenessQuantitative: vague entities according to their vagueness type.
  $\mathsf{VaguenessQualitative} \equiv \mathsf{VagueEntity} \sqcap \exists \mathsf{hasVaguenessType}.\{\mathsf{qualitative-vagueness}\}$
  $\mathsf{VaguenessQuantitative} \equiv \mathsf{VagueEntity} \sqcap \exists \mathsf{hhasVaguenessType}.\{\mathsf{quantitative-vagueness}\}$
- VaguenessDimension: Entities and their vagueness dimensions
  $\mathsf{VaguenessDimension} \equiv \mathsf{VagueEntity} \sqcap (\exists \mathsf{hasJustification.Justification}) \sqcap (\exists \mathsf{hasDimension.Dimension})$
- VaguenessDimensionInContext: Entities and their context-specific vagueness dimensions.
  $\mathsf{VaguenessDimensionInContext} \equiv \mathsf{VagueEntity} \sqcap (\exists \mathsf{hasJustification.Justification}) \sqcap (\exists \mathsf{hasDimensionInContext.DimensionInContext})$

Then, we have modified the above class descriptions in order to satisfy with Vagueness Rules that presented in section II-B.

- VR 3: The conjunction of a set of classes is vague if at least one of these classes is vague. The subclass of a vague class is also vague, unless this subclass is not vague entity.
  $\mathsf{VagueClass} \equiv (\mathsf{VagueEntity} \sqcap \mathsf{Class}) \sqcup (\exists \mathsf{subClassOf}.(\mathsf{VagueClass} \sqcap \neg \mathsf{NonVagueEntity}))$
- VR 4: The conjunction of a set of classes is quantitatively vague if all vague classes are quantitatively vague.
  $\mathsf{VaguenessQuantitative} \equiv \mathsf{VagueEntity} \sqcap \exists \mathsf{hasVaguenessType}.\{\mathsf{quantitative-vagueness}\} \sqcup \forall \mathsf{subClassOf.VaguenessQuantitative}$
- VR 5: The conjunction of a set of classes is qualitatively vague if at least one vague class is qualitatively vague.
  $\mathsf{VaguenessQualitative} \equiv \mathsf{VagueEntity} \sqcap \exists \mathsf{hasVaguenessType}.\{\mathsf{qualitative-vagueness}\} \sqcup \exists \mathsf{subClassOf.VaguenessQualitative}$
- VR 6: The quantitatively vague conjunction of classes is quantitatively vague in the super set of all the vague classes' dimensions.
  $\mathsf{VaguenessDimension} \equiv \mathsf{VagueEntity} \sqcap (\exists \mathsf{hasJustification.Justification}) \sqcap (\exists \mathsf{hasDimension.Dimension}) \sqcup (\forall \mathsf{subClassOf.VaguenessQuantitative})$
- VR 7: A vague property cannot be transitive.
  $\mathsf{VagueRole} \sqsubseteq \neg \mathsf{RoleTran}$

Although we could encode most of the vagueness constraints as OWL constraints, we still could not represent VR 8 as an OWL constraint due to the limitation of OWL 2 DL.

### *G. Syntactic Checking*

In many situations, one might want to define a class description with a conjunction of complex class descriptions. For example, $\mathsf{YoungPerson} \equiv \mathsf{Person} \sqcap \exists \mathsf{hasAge.Young}$ where $\mathsf{Young} \equiv \exists \mathsf{Age}.[Int \geq 15 \sqcap Int \leq 30]$.

We could represent the complex class description with an atomic class, but we still cannot identify whether the newly introduced class is vague or not.

Vagueness Syntactic Checking will apply to a class description that equivalent to a complex class description and a vague property that has its inverse (VR 8). This procedure is very straight forward because after the Compute procedure, we already have all vague entities. In addition, this operation will add all instances of VaguenessQuantitative. However, the syntactic checking process may introduce new vague entities which require to recalculating the knowledge base until there is no new axiom generated.

## IV. Evaluation

In this section, we present the early experimental result of our approach. Firstly, we test the metamodeling approach without syntactic checking with the test data. Secondly, we test the metamodeling approach with syntactic checking with the test data. Please note that we use HermiT Reasoner [2] as an external reasoner in our evaluation.

We have generated the test data to test our framework by utilizing the following pattern:

1) $A_1 \doteq B_1 \sqcap B_2 \sqcap B_3$ if $B_2$ is vague, then $A_1$ is vague (VR 3).
2) $A_2 \doteq B_{11} \sqcap B_{12} \sqcap B_{13}$ if $B_{11}$ is qualitative vague, then $A_2$ is also qualitative vague (VR 5).
3) $A_3 \doteq B_{21} \sqcap B_{22} \sqcap B_{23}$ if $B_{21}$, $B_{22}$ and $B_{23}$ are quantitative vague, then $A_3$ should be quantitative vague (VR 4).
4) $A_4 \doteq B_{31} \sqcap B_{32}$, $B_{31} \equiv \exists R_2.B_{32}$, $B_{33} \equiv \exists R_2.B_{34}$, $B_{32} \equiv B_{34}$, $A_5 \doteq B_{33} \sqcap B_{34}$, If $B_{33}$ is vague then, $A_4$, $A_5$ and $B_{31}$ also vague.
5) $R_1 \equiv \mathsf{inverse}(S_1)$. $S_1$ is vague then, $R_1$ is vague (VR 8).
6) $\mathsf{Transitive}(S_1)$ if $S_1$ is vague, then the knowledge base become inconsistent (VR 7).

The last pattern only added into the test data to test vagueness rule 7 (VR 7) otherwise, the reasoner will stop before we can get all results.

### A. Metamodeling Approach

| Entity | Expected | Result |
|---|---|---|
| VagueEntity | 22 | 20 |
| VagueClass | 14 | 14 |
| VagueProperty | 2 | 1 |
| VaguenessQualitative | 2 | 2 |
| VaguenesQuantitative | 4 | 3 |

Table I
NUMBER OF VAGUE ENTITIES FOR METAMODELING APPROACH

As we discuss in the previous section, without Syntactic checking enabled, we have missed two vague entities. One reason is that the OWL-based constraints for VaguenessQuantitative does not derive a new instance of the definition class because of the Open World Assumption. And another reason is we cannot represent VR 8 with DL constraint.

[2] http://hermit-reasoner.com/

### B. Metamodeling Approach with Syntactic checking

| Entity | Expected | Result |
|---|---|---|
| VagueEntity | 22 | 22 |
| VagueClass | 14 | 14 |
| VagueProperty | 2 | 2 |
| VaguenessQualitative | 2 | 2 |
| VaguenessQuantitative | 4 | 4 |

Table II
NUMBER OF VAGUE ENTITIES FOR METAMODELING APPROACH AND SYNTACTIC CHECKING

The results presented in the table II shows that the Metamodeling Approach with Syntactic checking could deliver more the complete result w.r.t the vagueness constraints.

## V. Related Work

### A. Higher-order DLs for Metamodeling Reasoning

One of the most popular approaches which is closely related to DLs is HiLog [6]. HiLog is logic with a higher-order syntax, which allows predicates to appear as arguments in atomic formula. Moreover, a satisfiable first-order formula without equality is also satisfiable under the HiLog semantics. The notion of reification of concepts [3] is proposed as a means to express meta-level classes, but the author does not address neither the issue of meta-roles, nor the issue of query answering.

Motik [14] and De Giacomo et al. [8] proposed higher-order extensions of DL under HiLog-style semantics. De Giacomo et al. have studied a stronger variant of the semantics where intensions are assigned also to complex entities.

Reasoning with metamodeling is also presented in [15], where the language OWL FA is proposed, which introduces a stratum number in class constructors and axioms to indicate the strata they belong to, and suitable constraints impose that TBox axioms are stated on classes of the same stratum, while ABox axioms can only involve elements of two consecutive strata.

Glimm and colleagues [9] proposed an approach to metamodel higher-order entities directly in SROIQ. However, this is not concerned complex construction and properties of a higher-order DL.

### B. Vagueness Reasoning

Vagueness in ontologies has been typically treated in the Semantic Web community by means of fuzzy logic and probabilistic techniques[13], [12], [4], [5], [18], [16], [7]. These approaches facilitate the definition of truth degrees for vague ontology elements and the reasoning they enable has to do with the way these degrees propagate across different elements. The vagueness Ontology, on the other hand, focuses on clarifying the intended meaning and interpretation of the vague entities such as the concept membership criteria of a given vague concept. For example, a fuzzy ontology may contain the statement *"John is an expert at ontologies to a degree of 0.8"* but the information on how the notion of expertise should be interpreted in the given domain or context is contained in the vagueness description of the property *"isExpertAt"*.

In the same sense, the kind of reasoning that we describe in this paper is not about inferring truth degrees, but rather vagueness descriptions; this makes our approach complementary to fuzzy ontology related works and it may be used to enhance the comprehensibility of fuzzy degrees. A fusion of the two levels, namely the degree-level and the vagueness description level, is an interesting future work.

## VI. Conclusions and Future Work

In this paper, we have extended the Vagueness Ontology by means of formal constraints and a reasoning framework that model the way vagueness descriptions of complex ontology elements can be derived from simpler ones. We have detailed a combined approach between OWL FA and the class-based approach by syntactically rewriting a domain ontology and integrate it with the Vagueness Ontology while preserving all semantics connections according to the extended FA semantics. Then, we detailed the reasoning approach to tackling the Vagueness Ontology's reasoning limitations. The early experimental results show that we could automatically derive the implicit vague entities from the explicit ones.

In the future, we would like to investigate about the completeness of the result set if we use an approximation based DL reasoner instead of sound and complete DL reasoner. Furthermore, we would like to investigate on how to use the tradition DL reasoner with the syntactic checking approach because our metamodeling approach may be too expensive in the large scale deployment.

## Acknowledgment

This work has been partially supported by the K-Drive project (FP7-286348).

## References

[1] Panos Alexopoulos, Silvio Peroni, Boris Villazón-Terrazas, Jeff Z. Pan, and José Manuél Gómez-Pérez. A metaontology for annotating ontology entities with vagueness descriptions. In *Proc. of URSW 2011-2013, Revised Selected Papers*, pages 100–121, 2014.

[2] Panos Alexopoulos, Boris Villazón-Terrazas, and Jeff Z. Pan. Towards vagueness-aware semantic data. In *Proc. of URSW 2013*, volume 1073, pages 40–45. CEUR-WS.org, 2013.

[3] Liviu Badea. Reifying concepts in description logics. In *Proc. of IJCAI 1997*, pages 142–149. Morgan Kaufmann, 1997.

[4] Fernando Bobillo and Umberto Straccia. fuzzydl: An expressive fuzzy description logic reasoner. In *FUZZ-IEEE*, pages 923–930. IEEE, 2008.

[5] Fernando Bobillo and Umberto Straccia. Fuzzy Ontology Representation using OWL 2. *International Journal of Approximate Reasoning*, 52(7):1073–1094, October 2011.

[6] Weidong Chen, Michael Kifer, and David Scott Warren. Hilog: A foundation for higher-order logic programming. *J. Log. Program.*, 15(3):187–230, 1993.

[7] Jhonatan Garcia, Jeff Z. Pan, Achille Fokoue, Katia Sycara, Yuqing Tang, and Federico Cerutti. Handling uncertainty: An extension of DL-Lite with Subjective Logic. In *Proc. of DL 2015*, 2015.

[8] Giuseppe De Giacomo, Maurizio Lenzerini, and Riccardo Rosati. On higher-order description logics. In *Proc. of DL 2009*, volume 477. CEUR-WS.org, 2009.

[9] Birte Glimm, Sebastian Rudolph, and Johanna Völker. Integrated metamodeling and diagnosis in owl 2. In *Proceedings of the 9th International Semantic Web Conference*, volume 6496 of *LNCS*, pages 257–272. Springer, November 2010.

[10] Pascal Hitzler, Markus Krötzsch, Bijan Parsia, Peter F. Patel-Schneider, and Sebastian Rudolph, editors. *OWL 2 Web Ontology Language: Primer*. W3C Recommendation, 27 October 2009. Available at http://www.w3.org/TR/owl2-primer/.

[11] Dominic. Hyde. *Vagueness, logic and ontology / Dominic Hyde*. Ashgate Aldershot, England ; Burlington, VT, 2008.

[12] Daphne Koller, Alon Levy, and Avi Pfeffer. P-classic: A tractable probabilistic description logic. In *Proceedings of AAAI-97*, pages 390–397, 1997.

[13] Thomas Lukasiewicz and Umberto Straccia. Managing uncertainty and vagueness in description logics for the semantic web. *Web Semant.*, 6(4):291–308, November 2008.

[14] Boris Motik. On the Properties of Metamodeling in OWL. *Journal of Logic and Computation*, 17(4):617–637, 2007.

[15] Jeff Z. Pan, Ian Horrocks, and Guus Schreiber. OWL FA: A Metamodeling Extension of OWL DL. In *Proc. of OWL-ED 2005*, 2005.

[16] Jeff Z. Pan, Edward Thomas, Yuan Ren, and Stuart Taylor. Tractable Fuzzy and Crisp Reasoning in Ontology Applications. In *IEEE Computational Intelligence Magazine*, 2012.

[17] S. Shapiro. *Vagueness in Context*. Oxford University Press, 2006.

[18] Giorgos Stoilos, Giorgos Stamou, Jeff Z. Pan, Vassilis Tzouvaras, and Ian Horrocks. Reasoning with Very Expressive Fuzzy Description Logics. *Journal of Artificial Intelligence Research*, 30:273–320, 2007.

[19] Kees. Van Deemter. *Not Exactly: In Praise of Vagueness*. Oxford University Press, 2010.