

# A MetaOntology for Annotating Ontology Entities with Vagueness Descriptions

Panos Alexopoulos<sup>1</sup>, Silvio Peroni<sup>2,3</sup>, Boris Villazon-Terrazas<sup>1</sup>, Jeff Z. Pan<sup>4</sup>,  
and Jose-Manuel Gomez-Perez<sup>1</sup>

<sup>1</sup> iSOCO, Spain,

{palexopoulos,bvillazon,jmgomez}@isoco.com

<sup>2</sup> Department of Computer Science and Engineering, University of Bologna, Italy

silvio.peroni@unibo.it

<sup>3</sup> STLab-ISTC, CNR, Italy

<sup>4</sup> Department of Computing Science, University of Aberdeen, UK

jeff.z.pan@abdn.ac.uk

**Abstract.** The emergence in the last years of initiatives like the Linked Open Data (LOD) has led to a significant increase in the amount of structured semantic data on the Web. Central role to this development has been played by ontologies, as these enable the representation of real world domains in an explicit and formal way and, thus, the production of commonly understood and shareable semantic data. Nevertheless, the shareability and wider reuse of such data can be hampered by the existence of vagueness within it, as this makes the data's meaning less explicit. With that in mind, in this paper we present and evaluate the Vagueness Ontology, a metaontology that enables the explicit identification and description of vague entities and their vagueness-related characteristics in ontologies. The rationale is that such descriptions, when accompanying vague ontologies, may narrow the possible interpretations that the latter's vague elements may assume by its users.

## 1 Introduction

Ontologies are formal shareable conceptualisations of domains, describing the meaning of domain aspects in a common, machine-processable form by means of concepts and their interrelations [10]. As such, their role in the Semantic Web is very important as they enable the production and sharing of structured data that can be commonly understood among human and software agents. To achieve this common understanding, one needs to ensure that the meaning of ontology elements is explicit and shareable. In other words, all their users have a clear, unambiguous and consensual understanding of what each ontological element actually represents. That's in fact the reason why, towards this goal, a number of relevant techniques and best practices have been proposed by the literature, such as for example the use of argumentation processes [34, 18] for consensus building on the structure and the content of an ontology. Despite these practices, however, a phenomenon that still affects, in a negative way, shareability and reusability of ontologies and semantic data is **vagueness**.

Vagueness is a common human knowledge and 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 [16, 30]. For example, some people are borderline tall: not clearly “*tall*” and not clearly “*not tall*”.

The potential and actual existence of vague terminology in ontologies and semantic datasets has already been identified by the community [35, 33, 21, 6, 2]. A characteristic group of such elements are categorisation relations where entities are assigned to categories with no clear applicability criteria. An example is the relation “*hasFilmGenre*”, found in LinkedMDB<sup>5</sup> and DBpedia<sup>6</sup>, that relates films with the genres they belong to. As most genres have no clear applicability criteria there will be films for which it is difficult to decide whether or not they belong to a given genre. A similar argument can be made for the DBpedia relations “*dbpedia-owl:ideology*” and “*dbpedia-owl:movement*”. Another group of vague elements comprises specialisations of concepts according to some vague property of them. Examples include “*Famous Person*” and “*Big Building*”, in the Cyc Ontology<sup>7</sup>, and “*Competitor*”, found in the Business Role Ontology<sup>8</sup>.

The important thing to notice in these examples is the lack of any further definitions that may clarify the intended meaning of the vague entities. For example, the definition of the concept “*Famous Person*” does not include the dimensions of fame according to which someone is judged as famous or not. This may lead to problematic situations.

More specifically, vague ontological definitions can cause **disagreements** among the people who develop, maintain or use it. Such a situation arose in a real life scenario where we faced significant difficulties in defining concepts like “*Critical System Process*” or “*Strategic Market Participant*” while trying to develop an electricity market ontology. When we asked our domain experts to provide exemplary instances of critical processes, there was dispute among them about whether certain processes qualified. Not only did different domain experts have different criteria of process criticality, but neither could anyone really decide which of those criteria were sufficient for the classification. In other words, the problem was the vagueness of the predicate “*critical*”.

While disagreements may be overcome by consensus, they are inevitable as more users alter, extend, or use ontologies. Imagine an enterprise ontology where the concept “*Strategic Client*” was initially created and populated by the company’s executive board, their implicit membership criterion being the amount of revenue the clients generate for the company. Imagine also the new R&D Director querying the instances of this concept while crafting an R&D strategy. If their own applicability criteria for the term “*Strategic*” do not coincide with the

<sup>5</sup> Available at <http://linkedmdb.org>

<sup>6</sup> Available at <http://dbpedia.org>

<sup>7</sup> Available at <http://www.cyc.com/platform/opencyc>

<sup>8</sup> Available at <http://www.ip-super.org>

board’s, using the returned list of clients might lead to poor decisions. Generalising these examples, some typical use-case scenarios where vagueness may be cause problems include:

1. **Structuring Data with a Vague Ontology:** When domain experts are asked to define instances of vague concepts and relations, then disagreements may occur on whether particular entities constitute instances of them.
2. **Utilising Vague Facts in Ontology-Based Systems:** When knowledge-based systems reason with vague facts, their output might not be optimal for those users who disagree with these facts.
3. **Integrating Vague Semantic Information:** When semantic data from several sources need to be merged then the merging of particular vague elements can lead to data that will not be valid for all its users.
4. **Evaluating Vague Semantic Datasets for Reuse:** When data practitioners need to decide whether a particular dataset is suitable for their needs, the existence of vague elements can make this decision harder. It can be quite difficult for them to assess *a priori* whether the data related to these elements are valid for their application context.

To reduce the negative effects of vagueness, we have put forward the notion of **vagueness-aware ontologies** [2], informally defined as “*ontologies whose vague elements 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 realise this. A more sophisticated example, as we will explain in subsequent sections, is the particular type of the entity’s vagueness or the applicability context of its definition. In all cases, our premise 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 elements may assume by human and software agents.

The focus of this paper is how vagueness-related metainformation may best be represented and applied to actual ontologies. For that, we describe here the **Vagueness Ontology (VO)**, an OWL metaontology that defines the necessary concepts, relations and attributes for creating explicit descriptions of vague ontology entities and (certain of) their characteristics. VO is meant to be used by both producers and consumers of ontologies; the former will utilise 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.

The motivation behind the development of VO is that, in our view, the vagueness-related metainformation should not be merely part of the ontology’s informal documentation, neither its representation can be facilitated by simply using OWL’s standard annotation properties such as `rdfs:comment`. The latter is because, as we will show in subsequent sections, one or more `rdfs:comment` values in an ontology entity cannot capture the more complex relations that exist between certain vagueness aspects.

The structure of the rest of the paper is as follows. In the next section we present related work while in section 3 we provide a detailed description of the Vagueness Ontology, including the requirements it is designed to cover, the conceptual elements (classes, relations etc.) it comprises and usage examples. In section 4 we present the results of a user-driven evaluation of the Vagueness Ontology, focusing on comprehensibility and usability aspects. Finally, in section 5 we cover some important discussion points regarding the benefits and current limitations of our approach, while in section 6 we summarise our work and outline its future directions.

## 2 Related Work

The practice of using ontologies for annotating various types of resources with meta-information has been exemplified by many works, including the *NLP Interchange Format (NIF)* [14], the *Extremely Annotational RDF Markup (EAR-MARK)* [4], and *Annotea* [17] for textual resources, as well as the more generic *Open Annotation Data Model (OADM)* [28] and *Provenance Ontology (PROV-O)* [19]. There are also several existing efforts for annotating ontologies. For general purpose ontology metadata we have Ontology Metadata Vocabulary (OMV) [13], Vocabulary of a Friend (VOAF)<sup>9</sup>. For metadata regarding ontology design and evolution there are the OWL 2 change ontology [24] and the Change and Annotations Ontology (CHAO) [23] as well as the C-ODO OWL metamodel for collaborative ontology design [12]. Finally, LexOMV [22] and Lemon [8] define metadata about multilinguality.

While the above vocabularies cover a large range of possible meta-information for ontologies, there is not yet, to the best of our knowledge, any specialised vocabulary for vagueness. The latter has so far been treated in the Semantic Web community mainly via fuzzy description logics, fuzzy ontologies and fuzzy query services [31, 25, 6], whose focus, however, is on enabling the definition and automated processing of fuzzy degrees of vague ontology entities and not so much on clarifying their intended interpretation (e.g. the concept membership criteria of a given vague concept). Thus, for example, a fuzzy ontology may contain the statement “*John is expert at ontologies to a degree of 0.8*” but there is no information on how the notion of expertise should be interpreted in the given domain or context. Therefore, as it will become clear in the rest of the paper, our approach is complementary to fuzzy ontology related works and it may be used to enhance the comprehensibility of fuzzy degrees.

## 3 The Vagueness Ontology

The Vagueness Ontology<sup>10</sup> has been developed following the *SAMOD*<sup>11</sup> (*Simplified Agile Methodology for Ontology Development*) methodology and its relevant

<sup>9</sup> Available at <http://lov.okfn.org/vocab/voaf/v2.1/index.html>

<sup>10</sup> Available at <http://www.essepuntato.it/2013/10/vagueness>

<sup>11</sup> Available at <http://www.essepuntato.it/samod>

documentation is available online<sup>12</sup>. In this section, we focus on describing the requirements the ontology has been designed to satisfy and the main elements it consists of.

### 3.1 Vagueness Ontology Requirements

In an ontology, vagueness may primarily appear in the definitions of classes, object and datatype properties, and datatypes. A class is vague if, in the given domain, context or application scenario, it admits borderline cases, namely if there are (or could be) individuals for which it is indeterminate whether they instantiate the class. Typical vague classes are attributions, namely classes that reflect qualitative states of entities (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 they 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. An example is the datatype *RestaurantPriceRange* when this comprises the terms *cheap*, *moderate* and *expensive*.

The Vagueness Ontology should enable the annotation of an ontological entity (class, relation or datatype) with a description of the nature and characteristic of its vagueness. In particular, the first thing such a description should explicitly state is whether the entity is actually vague or not. For example, the ontology class *StrategicClient* defined as *A client that has a high value for the company* is (and should be annotated as) vague while the definition of *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 element can have a non-vague definition (e.g. *TallPerson* when defined as *A person whose height is at least 180cm*). Then this element is not vague in the given ontology and that is something that needs to be explicitly stated.

The second important vagueness characteristic to be explicitly represented is its type. Vagueness can be described according to at least two complementary types: quantitative (or degree) vagueness and qualitative (or combinatory) vagueness [16]. A predicate has degree-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 quantity while *red* can be vague for both brightness and saturation). A predicate has combinatory vagueness if there are 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. A classical example of this type is *religion* as there are certain features that all religions share (e.g. beliefs in supernatural beings, ritual acts) yet it is not clear which are able to classify something as a religion. Based on this typology, we suggest that for a given vague entity it is important to represent and share the following explicitly:

<sup>12</sup> Available at <http://www.essepuntato.it/2013/10/vagueness/documentation>

- **The type of the entity’s vagueness:** Knowing whether an entity has quantitative or qualitative vagueness is important as elements with an intended (but not explicitly stated) quantitative vagueness can be considered by others as having qualitative vagueness and vice versa. Assume, for example, that a company’s CEO does not make explicit that for a client to be classified as strategic, the amount of its R&D budget should be the only factor to be considered. Then, even though according to the CEO the vague class “*StrategicClient*” has quantitative vagueness in the dimension of the R&D budget amount, it will be hard for other company members to share the same view as this term has typically qualitative vagueness.
- **The dimensions of the term’s quantitative vagueness:** When the entity has quantitative vagueness it is important to state explicitly its intended dimensions. E.g., if a CEO does not make explicit that for a client to be classified as strategic, its R&D budget should be the only pertinent factor, it will be rare for other company members to share the same view as the vagueness of the term “*strategic*” is multi-dimensional.

Furthermore, vagueness is **subjective** and **context dependent**. The first has to do with the same vague entity being interpreted differently by different users. For example, two company executives might have different criteria for the entity “*StrategicClient*”, the one the amount of revenue this client has generated and the other the market in which it operates. Similarly, context dependence has to do with the same vague entity being interpreted or applied differently in different contexts even by the same user; hiring a researcher in industry is different to hiring one in academia when it comes to judging his/her expertise and experience.

Therefore we additionally suggest that one should explicitly represent the **creator** of a vagueness annotation of a certain entity as well as the **applicability context** for which the entity is defined or in which it is used in a vague way. In particular, 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). Please note that here we adopt the context-as-a-box metaphor [5] according to which a context is a “box” that contains knowledge in form of logical statements and whose boundaries are determined by specific contextual attributes (e.g. location, time, purpose etc). When a vague term is related to a particular context, then this context has the jurisdiction to interpret the term’s meaning and assess its validity in given statements [3].

Summarising the above, the Vagueness Ontology should enable users to ask the following competency questions about the entities of an ontology:

- *What entities have been explicitly defined either as vague or non-vague?*
- *What entities that have been defined both as vague and non-vague at the same time and why?*
- *What entities of a specific type (e.g., classes) have been defined either as vague or non-vague?*

- What entities are characterised by a specific vagueness type?
- What entities have been recognised as vague, by whom and according to which vagueness type (if any)?
- What entities have quantitative vagueness and in what dimensions?
- What entities have quantitative vagueness, in what dimensions and what is the context of their dimensions (if any)?
- What entities are vague, in what contexts and according to whom?

### 3.2 Ontology Anatomy

An overall view of the Vagueness Ontology (VO) is depicted in Figure 1 via a Graffoo diagram [11] that describes its main classes and properties. VO uses several entities defined in external ontologies, i.e., the PROV-O [19] (prefix *prov*), OADM [28] (prefix *oa*), and the Situation ontology design pattern<sup>13</sup> (prefix *sit*). To show how to use the various entities of the ontology to describe vagueness/non-vagueness annotations, we introduce the following natural language scenario:

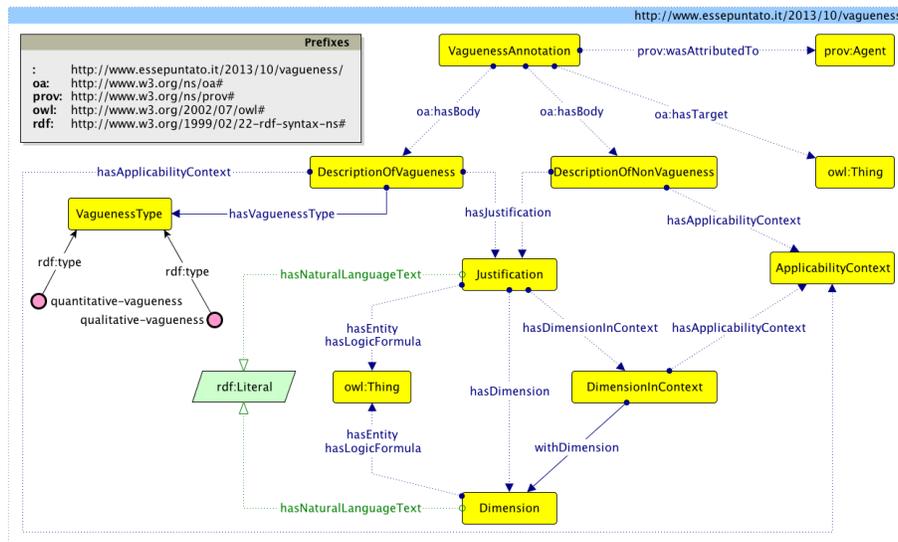


Fig. 1. The Graffoo diagram of the overall structure of the Vagueness Ontology.

The object property *ex:isExpertInResearchArea* is considered vague by John Doe in the context of researcher hiring. Moreover, he describes it as quantitatively vague since, for him, expertise is relevant to the number

<sup>13</sup> Available at <http://www.ontologydesignpatterns.org/cp/owl/situation.owl>.

of her publications and projects; two different dimensions that he thinks relates to the contexts of Academia (i.e., number of relevant publications) and Industry (i.e., number of relevant projects).

The main class of the ontology is *VaguenessAnnotation*, which describes any annotation (i.e., *oa:Annotation*) of an ontological entity with information about its vagueness. A vagueness annotation is a particular act done by someone (i.e., an agent, identified by an individual of the class *prov:Agent*) who associates a description of vagueness/non-vagueness (called the body of the annotation, and defined through the property *oa:hasBody*) to the entity in consideration (called the target of the annotation, and defined through the property *oa:hasTarget*). This is formalised as follows<sup>14</sup>:

```
Class: VaguenessAnnotation
  SubClassOf: prov:Entity that prov:wasAttributedTo some prov:Agent,
             oa:Annotation that (oa:hasTarget exactly 1) and
             (oa:hasBody min 1
              (DescriptionOfNonVagueness or DescriptionOfVagueness))
```

Considering the aforementioned example, the annotation made by John Doe can be expressed as follows:

```
@prefix : <http://www.essepuntato.it/2013/10/vagueness/> .
@prefix ex: <http://www.essepuntato.it/resource/> .
ex:annotation a :VaguenessAnnotation ; prov:wasAttributedTo ex:john-doe ;
  oa:hasBody ex:description-of-vagueness ;
  oa:hasTarget ex:isExpertInResearchArea .
ex:isExpertInResearchArea a owl:ObjectProperty .
ex:john-doe a prov:Agent .
```

A vagueness annotation must specify a description of vagueness or non-vagueness for the annotated entity, in the form of an instance of the class *DescriptionOfVagueness* or *DescriptionOfNonVagueness* respectively. Vagueness descriptions must specify a vagueness type (one of the individuals of the class *VaguenessType*, i.e., *quantitative-vagueness* and *qualitative-vagueness*), and must provide at least one justification (i.e., an individual of the class *Justification*) for considering the target ontological entity vague. The individuals of the class *DescriptionOfNonVagueness*, instead, require only the specification of at least one justification. This class is meant to be used for entities that would typically be considered vague but which, for some reason, in the particular ontology are not (e.g. the “*TallPerson*” example in section 3.1). Formalisation here is as follows:

```
Class: DescriptionOfNonVagueness SubClassOf: hasJustification min 1
Class: DescriptionOfVagueness
  SubClassOf: hasJustification min 1 , hasVaguenessType exactly 1
ObjectProperty: hasJustification
  Domain: DescriptionOfNonVagueness or DescriptionOfVagueness
  Range: Justification
ObjectProperty: hasVaguenessType
  Domain: DescriptionOfVagueness Range: VaguenessType
```

Considering again the previous example, the John Doe’s description of vagueness can be defined as follows:

<sup>14</sup> All the entities of the Vagueness Ontology are introduced in Manchester Syntax [15], while the examples of use of the ontology are presented in Turtle [27].

```

ex:description-of-vagueness a :DescriptionOfVagueness ;
:hasJustification ex:justification ;
:hasVaguenessType :quantitative-vagueness .

```

The justifications of descriptions of vagueness/non-vagueness (i.e., individuals of the class *Justification*) aim at explaining the possible reasons behind such descriptions. Vagueness dimensions, in turn, (i.e., individuals of the class *Dimension* referred by the object property *hasDimension* and being always part of a justification) always refer to descriptions of quantitative vagueness and indicate some measurable characteristic of the annotated entity in which it is vague. Both justifications and dimensions can be defined as natural language text (i.e., the data property *hasNaturalLanguageText*), an entity (i.e. the object property *hasEntity*), a more complex logic formula (i.e., the object property *hasLogicFormula*) or any combination of them. The relevant formalisation is as follows:

```

Class: Justification
  SubClassOf: hasNaturalLanguageText some rdfs:Literal or
             hasEntity some owl:Thing or hasLogicFormula some owl:Thing
ObjectProperty: hasDimension
  Domain: Justification that inverse hasJustification only
         (DescriptionOfVagueness that
          (hasVaguenessType value quantitative-vagueness))
  Range: Dimension
Class: Dimension
  SubClassOf: hasNaturalLanguageText some rdfs:Literal or
             hasEntity some owl:Thing or hasLogicFormula some owl:Thing

```

Please note that while the properties *hasEntity* and *hasLogicFormula* share the same range class, i.e., *owl:Thing*, their intended meaning is different. The former property can be used to specify a certain resource (e.g., *dbpedia:H-index*) as (part of) a justification of a certain description. The latter property, instead, is used to link to a resource, which provides a justification, that actually “puns” a particular restriction or constraint on certain entities, e.g., *ex:hasNumberOfPublication some integer[>0]*.

Continuing the previous example, the justification and the related dimensions can be described as follows:

```

ex:justification a :Justification ;
:hasNaturalLanguageText "It is not possible to define the exact minimum
  number of relevant publications and projects that make a a researcher
  expert in a given area." ;
:hasDimension ex:dimension-publications , ex:dimension-projects .
ex:dimension-publications a :Dimension ;
:hasNaturalLanguageText "The number of relevant publications." .
ex:dimension-projects a :Dimension ;
:hasNaturalLanguageText "The number of relevant projects." .

```

As introduced before, descriptions of vagueness/non-vagueness and related dimensions can be characterised by particular contexts of application (i.e., individuals of the class *ApplicabilityContext*), which means that they can be applied within the boundaries of such particular contexts (i.e., the same entity can be vague in one context and non-vague in another). The contextualisation of descriptions is facilitated by an assertion between the description in consideration and the related context through the object property *hasApplicabilityContext*. In the case of dimensions, on the other hand, the context-dependent object is the *relation* between justifications and dimensions. Thus, to represent this, we reify

the relation linking a justification to a dimension using an instance of the class *DimensionInContext*, that allows one to specify and the applicability context of such relation. VO formalises this as follows:

```

ObjectProperty: hasApplicabilityContext
  Domain: DescriptionOfNonVagueness or DescriptionOfVagueness or
         DimensionInContext
  Range: ApplicabilityContext
ObjectProperty: hasDimensionInContext
  Domain: Justification that inverse hasJustification only
         (DescriptionOfVagueness that
          (hasVaguenessType value quantitative-vagueness))
  Range: DimensionInContext
Class: DimensionInContext
  SubClassOf: sit:Situation , withDimension exactly 1 ,
             hasApplicabilityContext exactly 1
ObjectProperty: withDimension
  Characteristics: Functional SubPropertyOf: sit:isSettingFor
  Domain: DimensionInContext Range: Dimension

```

According to the above definitions, it is possible to complete the description of the aforementioned example as follows:

```

ex:description-of-vagueness
  :hasApplicabilityContext ex:researcher-hiring-context .
ex:researcher-hiring-context a :ApplicabilityContext .
ex:justification :hasDimensionInContext
  ex:dimension-publications-in-context , ex:dimension-projects-in-context .
ex:dimension-publications-in-context a :DimensionInContext ;
  :withDimension ex:dimension-publications ;
  :hasApplicabilityContext ex:academia-context .
ex:dimension-projects-in-context a :DimensionInContext ;
  :withDimension ex:dimension-projects ;
  :hasApplicabilityContext ex:industry-context .
ex:academia-context a :ApplicabilityContext .
ex:industry-context a :ApplicabilityContext .

```

This approach allows the reuse of the same dimension in different contexts and reasoners to infer automatically all the *hasDimension* assertions starting from the individuals of the class *DimensionInContext* by means of the sub-property chain *hasDimensionInContext o withDimension* defined in the object property *hasDimension*.

## 4 Vagueness Ontology Evaluation

As an initial assessment of VO's correctness, we used the online tool OOPS!<sup>15</sup> [26] to detect potential modelling errors; results indicated no critical errors. Beyond that, we asked from a group of people with a working knowledge of ontologies to use VO to query ontologies that were already annotated with vagueness descriptions. Our goal was to evaluate the comprehensibility and usability of the current version of the ontology and get feedback.

The term "usability" here denotes the easiness by which a user of an ontology that has already been annotated with VO, can access (via SPARQL) and understand this vagueness-related meta-information. To assess this kind of usability, we asked subjects to study VO starting from its sources, documentation and

<sup>15</sup> Available at <http://oeg-lia3.dia.fi.upm.es/oops/index-content.jsp>

additional material we provided, as well as to use a SPARQL endpoint in order to answer specific competency questions regarding the vagueness of a concrete VO-annotated ontology. The usability of VO in terms of the easiness by which an ontology engineer can annotate vague ontologies is going to be evaluated in future work.

#### 4.1 Experimental Setting

We asked 22 subjects to perform three unsupervised tasks involving querying a SPARQL endpoint containing vagueness information about four entities, three classes and one object property. There were no “administrators” observing the subjects while they were undertaking these tasks, and we made sure that none of the subjects was previously aware of VO. In the end, 10 of these subjects completed the tasks. However, only 6 of which had enough experience in performing proper SPARQL queries, which is a mandatory requirement that subjects had to demonstrate in order to use quantitative data for assessing users’ performance in addressing the tasks. Therefore, we used all the 10 subjects’ data for analysing the usability of VO as gathered through the questionnaires introduced below, while we considered only the results related to the SPARQL-aware users for evaluating quantitative outcomes

More specifically, the assessment of the actual subject’s experience concerning the ability to provide appropriate SPARQL queries was derived from the answers the subject provided in a preliminary questionnaire, composed by self-assessment questions about subject’s preliminary knowledge. In addition, we also analysed the actual SPARQL queries made by the subject during the test, in order to understand if (s)he was able to use basic SPARQL constructs such as *UNION* and *OPTIONAL*, that were necessary for addressing the tasks we proposed properly. In case these requirements were not satisfied, we did not consider the subject’s SPARQL queries in the quantitative analysis of such data, in order not to bias the results. Therefore, we used only the queries provided by 6 out of 10 subjects for our quantitative analysis.

On the other hand, we thought that the understandability/learnability of the ontology could be assessed considering all the 10 subjects, since these aspects refer to the subjective perception of people when understanding the ontology and querying ontological data. During the test, we did not tell subjects whether their SPARQL queries were right or not and, thus, the actual correctness of such queries did not bias the subjects’ personal perception of the ontology.

In all cases, the ontology we used contained nine annotations, seven of which pointed to descriptions of vagueness (one of those had an applicability context specified), while the remaining two referred to descriptions of non-vagueness (one of those had an applicability context specified). Some of these descriptions referred to seven justifications, while two of these justifications were linked to two dimensions each (in two cases, the justification-dimension relation presented a particular applicability context). The tasks given to the subjects involved the latter translating a natural language query into SPARQL. These queries were designed to ensure that subjects had to use all the entities of VO so as to reach

a solution. Both the dataset and the tasks were based on the examples and the informal competency questions we had produced during the development of VO.

**Table 1.** The three natural language questions of each task (T1, T2 and T3) to translate in SPARQL.

<b>T1</b>	What are all the entities that have been defined either as vague or non-vague and why?
<b>T2</b>	What are all the entities, their related vagueness type, and their related OWL type that have been defined vague by someone?
<b>T3</b>	What entities have quantitative vagueness, in what dimensions and what is the context of their dimensions?

The evaluation session was structured as follows. We first asked subjects to complete a short multiple-choice questionnaire about their background knowledge and skills in OWL, ontology engineering, SPARQL, PROV-O and OADM (max. 2 mins.). Then, we asked subjects to study VO (max. 25 mins.), providing them the ontology source in RDF/XML, the complete online documentation with the diagram of Figure 1, and usage examples. Then, we asked them to complete the three tasks listed in Table 1 (max. 15 mins.), allowing them to test the SPARQL translations on the dataset, available as SPARQL endpoint. During that, no access to the any exemplar SPARQL queries was given. Finally, we asked subjects to fill in two short questionnaires, one multiple choice and the other textual, to report their experience of using VO (and its related material) to complete these tasks (max. 5 min). All the questionnaires<sup>16</sup> and all the outcomes of the experiments<sup>17</sup> are available online.

## 4.2 Evaluation

Out of the 30 tasks in total (3 tasks given to each of 10 subjects), 9 were completed successfully (i.e., the right SPARQL queries were given), while 9 had incorrect answers or were not completed at all, giving an overall success rate of 50%. The remaining 12 ones were not considered in this quantitative analysis since the related 4 users had proved to have not enough experience in performing SPARQL queries. The 9 successes were distributed as follows: 2 (out of 6) in Task 1, 6 in Task 2, and 1 in Task 3. A similar analysis can be done for the actual rows of the 6 users' outcomes matching with the expected results. In this case, we compared the each row returned by executing each user's SPARQL query with the expected rows, listing all the true positives (tp), false positives (fp), and false negatives (fn). We calculated the overall average precision (P) (i.e.,  $\text{tp}/(\text{tp}+\text{fp})$ ) and average recall (R) (i.e.,  $\text{tp}/(\text{tp}+\text{fn})$ ), calculated by considering those obtained by each subject, and we obtained  $P = 0.61$  and  $R = 0.75$ .

<sup>16</sup> Available at <http://esurv.org?u=vagueness-ontology>.

<sup>17</sup> Available at <http://www.essepuntato.it/2013/10/vagueness/evaluation>.

The average precision and recall for each task were  $P = 0.49$  and  $R = 0.44$  in Task 1,  $P = 1$  and  $R = 1$  in Task 2, and  $P = 0.66$  and  $R = 0.83$  in Task 3.

As shown by these quantitative results, the second task was always answered correctly, while issues arose when trying to answer to tasks 1 and 3. On the one hand, in Task 1 we think two users (out of three who provided wrong answers) simply made syntactic mistakes (i.e., one returns the annotation individuals instead of the kinds of descriptions linked by such annotations, while the other named two SPARQL variables in the same way), which could be due to a rushed reading of the task or a distraction when writing the SPARQL query. On the other hand, in Task 3 it seems that subjects' mistakes related to a partial understanding of the ontology, since five of them provided imprecise solutions to the task. This seemed to depend on the possibility of describing dimensions involved in descriptions of quantitative vagueness as contextual objects or not, as introduced in Section 3.2. Although we were aware of possible misinterpretation of such part of the ontology, we decided to define dimensions by using the same pattern proposed in PROV-O, where certain relations, for instance between an entity and an agent (e.g., *prov:wasAttributedTo*), can be qualified, if needed, by reifying them as proper classes (e.g., *prov:Attribution*) linking to the entity and the agent in consideration. Of course, in all the above, one needs to consider the constrained time that participants had to study the ontology and perform the tasks.

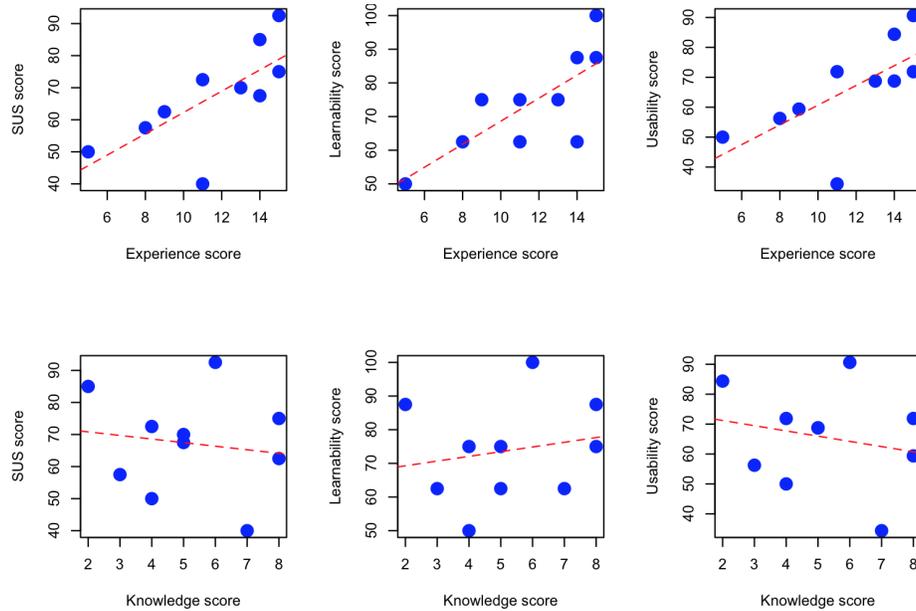
The usability score for VO (considered together with its documentation and examples) was computed using the *System Usability Scale (SUS)* [7], by using the answers provided by all the 10 users. SUS is a well-known questionnaire used for the perception of the usability of a system, and it has been already used in the past for assessing the usability of ontologies (cf. [9]). SUS has the advantage of being technology independent (it has been tested on hardware, software, Web sites, etc.) and it is reliable even with a very small sample size [29]<sup>18</sup>. In addition to the main SUS scale, we also were interested in examining the sub-scales of pure *Usability* and pure *Learnability* of VO, as proposed recently by Lewis and Sauro [20]. The mean SUS score for VO was 67.3 (in a 0 to 100 range), approaching the target score of 68 to demonstrate a good level of usability [29]. The mean values for the SUS sub-scales Usability and Learnability were 68.8 and 73.4 respectively.

In addition, two sub-scores were calculated for each subject by considering the values of the answers given in the background questionnaire (according to

---

<sup>18</sup> Even if confidence intervals of the SUS scores will be rather wide (e.g., in our experiment we obtained [56.06, 78.45]), the average SUS score will be surprisingly stable even with a small sample. As stated in [29] and summarised in his blog (see <http://www.measuringusability.com/blog/10-things-SUS.php> for more details), Sauro “did several computer simulations and showed that [...] the mean from a sample size of just 5 repeated 1000 times [...] was within 6 points of the true SUS score” in the 50% of the 1000 samples used – note that the true SUS score was calculated using the original big sample Sauro had available. This means that “you get within the ballpark of the actual SUS score in more than half of the cases with very small sample sizes” – e.g., “if the actual SUS score was a 74, average SUS scores from five users will fall between 68 and 80 half of the time”.

a 0 to 4 value range for each question). The first sub-score – composed of five questions and, thus, ranging from 0 to 20 – concerned the subject’s *experience* with (the development of) ontologies. The other sub-score – composed of three questions and, thus, ranging from 0 to 12 – concerned the subject’s personal *knowledge* about SPARQL, PROV-O and OADM. As shown in Figure 2, we have plotted these subject’s sub-scores (x-axis) with the subject’s SUS value and the other sub-scales (y-axis) – and we have also included red dashed lines referring to the related *Least Squares Regression Lines*. Even if we cannot have any statistical significance of such comparisons because of the small size of our sample, it seems that the plots suggest some sort of positive correlation between the experience sub-scores and the SUS values – i.e., the more a subject knew about ontologies in general, the more VO is perceived as usable. The plots referring to the other aspect, namely the relation between the knowledge sub-scores and the SUS values, does not seem to provide enough evidence to speculate on any sort of correlation.



**Fig. 2.** Six plots showing the relation between subjects’ *experience* and *knowledge* scores and the related subjects’ SUS values and the other sub-scales. The red dashed lines were calculated by using the Least Squares Regression method.

Axial coding of the personal comments expressed in the final questionnaires [32] revealed a small number of perceived issues. Only 8 of all the subjects tested provided meaningful comments that were used for the study, and, of the 7 terms that were identified as significant in the comments, only 5 (4 positive and 1

negative) were mentioned by more than two individuals (albeit sometimes with different words), as shown in Table 2. The only negative issue mentioned by more than two subjects, i.e., the ambiguities in some ontological terms, was mainly highlighted by the subjects whom answers to tasks were not considered in the quantitative evaluation due to their inexperience in SPARQL. This seems to suggest some sort of (cor)relation between the understandability of VO and the experience users had in using SPARQL.

**Table 2.** Terms – four positive (+) and one negative (–) – mentioned by more than two individuals in the final questionnaire responses.

Term	Description	Frequency
HTML documentation (+)	The documentation of VO was suggested as one of the success features to help users understanding the ontology and, thus, to consult for writing the SPARQL queries.	4 out of 8
ABox examples (+)	Similarly to the documentation, the snippets provided to show how to use VO for describing vagueness entities were a crucial aspect for writing SPARQL queries, since they were used to understand the intended usage of VO entities.	3 out of 8
SPARQL endpoint (+)	Although it was not one of the targets of the evaluation, the SPARQL endpoint we provided to test the SPARQL queries was considered useful to assess their correctness and to browse the dataset.	3 out of 8
Grafoo diagram (+)	The diagram in Figure 1, which is part of the ontology documentation, received explicit mention in comments as an effective way to quickly understand the relation among classes.	3 out of 8
Entity ambiguities (–)	Users found some ontological entities, e.g., <i>context</i> and <i>dimension</i> , quite ambiguous since they could be interpreted in different ways. In addition, the possibility of providing different descriptions of vagueness for a certain concept had been perceived as a drawback, as well as the fact that one needs to deal with several levels of indirection in order to express precisely the vagueness of concepts.	3 out of 8

## 5 Discussion Points

### 5.1 Benefits of Consuming Vagueness-Aware Ontologies

The Vagueness Ontology is to be used by producers and consumers of ontologies and semantic datasets, so as to create and consume vagueness ontology descriptions respectively. Regarding the consumption of a vagueness-aware ontology, a first benefit of a vagueness-aware ontology for potential users is that it makes them aware of the existence of vagueness by explicitly stating the vague elements. This is important as vagueness is not always obvious to people (and certainly never to systems), meaning that it can be easily overlooked and lead to the negative effects described in previous sections. A second benefit is that it enables users to query each of the vague element’s metainformation (vagueness dimensions, applicability context etc) and use it in order to reduce these effects.

To show how this may be done let’s revisit the four scenarios of section 1. In the first scenario, involving the structuring of data with an existing vague ontology, the problem is that disagreements may occur on whether particular objects are actually instances or not of vague concepts or relations. If, however, information like the dimensions and applicability conditions and contexts of these elements are made known to the people who perform this task, then the possible interpretation space of them will be reduced. For example, if it is known that in order to classify a given company as a competitor, one needs to consider only the number of common business areas target markets, then other possible dimensions (e.g. the geographical proximity) will be excluded. This exclusion should reduce the number of potential disagreements.

In the second scenario, where vague ontological elements are utilised within some end-user application, the availability of vagueness metainformation can help the system’s developers in two ways. First, it will make them aware of the fact that the ontology contains vague information and thus some of the system’s output might not be considered accurate by the end-users. Second, they may use the vagueness metainformation to try to deal with that fact. For example, in a recommendation scenario, the applicability context of a vague axiom can be used as part of an explanation to the user of why a particular item was recommended. That might not change the user’s opinion on whether this recommendation is accurate, but the potential user’s feedback could help pin down the particular element’s vagueness as the cause of this inaccuracy and take appropriate action.

In the third scenario, when two or more ontologies need to be integrated, the vagueness metamodel can be used to compare the “compatibility” of these ontologies in terms of vagueness. For example, if the same two vague classes have different vagueness dimensions (e.g. the vague class “Strategic Client”), then the one class’s set of instance membership axioms might not be appropriate for the second one’s as it might have been defined under a different interpretation of the class’s vagueness. A simple query to the two ontologies’ vagueness metamodel could reveal this issue. Similarly, in the case of evaluating given ontologies and semantic datasets for reuse purposes, the metamodel can be used to compare the vagueness compatibility of the dataset with the intended domain and application scenario. Table 3 summarises the above use case scenarios and the way the metamodel may be used and benefit each of them.

From the above, it is evident that if the vagueness characteristics that VO specifies (dimensions, context, etc.) were merely part of its documentation and not explicitly represented as metadata, this kind of querying would not be possible. Moreover, as the VO captures formally the relations that exist between these characteristics (e.g., the relation between a dimension and a context), the same kind of querying would not be possible if these relations were defined using merely the `rdfs:comment` annotation property of OWL. In such a case, if the ontology user would like, for example, to measure the number of different dimensions and contexts a particular entity is vague in, he/she would have to parse, via some NLP method, the entity’s `rdfs:comment` value; a process obviously not very effective or easy to perform. On the other hand, with VO as a basis one

**Table 3.** Vagueness Metamodel Usage and Benefits in Different Scenarios.

Use Case Scenario	Metamodel Usage	Expected Benefits
Structuring data with a vague ontology	Communicate the meaning of the vague elements to the domain experts.	Make the job of the experts easier and faster and reduce disagreements among them.
	Use the metamodel to characterise the created data’s vagueness.	Enhance the future usability and shareability of the data.
Utilising vague semantic data in an ontology-based system.	Check which data is vague	Know a priori which data may affect the system’s effectiveness.
	Use the properties of the vague elements to provide vagueness-related explanations to the users.	
Integrating Vague Semantic Datasets	Compare same-name vague elements across datasets according to their vagueness type and dimensions	Avoid integrating incompatible elements.
Evaluating a Vague Semantic Dataset for Reuse	Query the metamodel to check the vagueness compatibility of the dataset with the intended domain and application scenario.	Avoid re-using (parts of) datasets that are not compatible to own interpretation of vagueness.

can access an ontology’s vagueness-related metainformation via SPARQL and, potentially, via more high-level services that are suitable not only to ontology engineers but also to domain experts, application developers and data analysts.

## 5.2 Creating Vagueness-Aware Ontologies

Annotating ontologies with VO is currently a manual task, with knowledge engineers and domain experts having to detect the vague elements, determine the relevant characteristics (type, dimensions, etc.) and instantiate VO. How this may be best facilitated is out of this paper’s scope but it is an important aspect of our ongoing and future work. An example of this work is a system we’ve developed that is able to automatically detect ontology elements that are potentially vague [1]. The system uses a binary classifier that may distinguish between vague and non-vague term word senses and, consequently, between vague and non-vague linguistic definitions of ontology entities. Thus, for example, the definition of the ontology class “*StrategicClient*” as “*A client that has a high value for the company*” would be classified as vague while the definition of “*AmericanCompany*” as “*A company that has legal status in the Unites States*” would not. Our goal is to incorporate this classification functionality into an ontology authoring tool that will take as input an ontology, detect automatically its vague entities, guide the user into annotating them with the Vagueness Ontology in a Q&A manner and give as output a vagueness annotation for the given ontology.

## 5.3 Reasoning with Vagueness-Aware Ontologies

The current version of VO has not been developed with automated reasoning in mind, primarily because we have not yet analysed vagueness in adequate

depth so as to define more complex axioms that may facilitate some kind of reasoning. Moreover, some of VO’s information, such as dimensions or contexts, is currently described in a textual (and thus imprecise) way, thus making it harder to perform very detailed reasoning. Both these limitations have been purposefully not tackled in this first version of VO, in order to avoid an increased complexity that could discourage people from adopting it and start using it to annotate their ontologies.

In principle, reasoning with VO can be made possible by defining constraints and inference rules that determine how vagueness and its characteristics proliferate when defining more complex OWL axioms, such as complex classes or subsumption relations. A simple example is to say that *“The conjunction of a set of classes is quantitatively vague if all (vague) classes are quantitatively vague whereas it is qualitatively vague if at least one (vague) class is qualitatively vague”*. Then, a vagueness (meta-)reasoner could infer a conjunctive class’s vagueness type by considering the types of its constituent classes. Similarly, one could say that *“The inverse of a vague property has the same vagueness characteristics (type, dimensions, contexts, etc.) as the original property”*. On the other hand, it is a matter of further analysis whether and in what way a class’s vagueness’s characteristics are “transferred” to its subclasses. Such an analysis, that will try to identify and implement a comprehensive set of valid reasoning rules for VO, is left as future work.

As far as the imprecise nature of VO’s textual content is concerned, its potentially inhibitive role in reasoning depends on the particular reasoning rule at hand. For example, the rule in the above paragraph regarding the vagueness type of a conjunctive class is not really affected by imprecision. On the other hand, the implementation of a rule such as *“The conjunction of a set of quantitatively vague classes is quantitatively vague in the superset of all these classes’ dimensions”* would require the comparison of vagueness dimensions (and probably contexts) which, when represented as simple strings, can be imprecise. For such cases, a more formal representation of dimensions and contexts (with, e.g., taxonomical relations between contexts) would be probably necessary; nevertheless, such a representation needs to be contemplated along with the specification of VO’s reasoning behaviour and, for that, is left as future work.

## 6 Conclusions and Future Work

In this paper we presented and evaluated the Vagueness Ontology (VO), a metaontology for annotating vague ontological entities with descriptions that describe the nature and characteristics of their vagueness in an explicit way. The metaontology is meant to be used by both producers and consumers of ontologies and semantic datasets, with the former utilising it to annotate the vague part of their produced ontologies and the latter querying this metainformation in order to make a better use of them.

VO’s high-level goal is to raise the awareness of human producers and consumers of ontologies and semantic data about vagueness and the potential prob-

lems it may cause, and provide them with the means to produce/consume ontologies with a clearer meaning. At the moment, there are neither established practices nor tools in the Semantic Web community for working with vagueness, the result being that vague ontologies and semantic data are created and used without realising the meaning explicitness issues that may arise. Moreover, it should be made clear that our work does not aim to “get rid” of vagueness; on the contrary, we want to highlight it as a central issue in the development of the Semantic Web and, for the scenarios we have identified in this paper, make it more manageable and less problematic by making it explicit, not eliminating it.

Regarding VO’s evaluation, our goal was to evaluate how (ontology-savvy) users understood VO. For that, we performed a time-constrained, user-based evaluation of VO showed a satisfying level of clarity and usability. Future experiments will involve domain experts and ontology engineers using VO to annotate ontologies; these experiments, however, will be performed when we have developed appropriate tooling for using VO.

This development will form part of our future work, aiming towards facilitating the easier and seamless usage of VO for the production of vagueness-aware ontologies, not only by ontology engineers but also by domain experts, application developers and data analysts. For that, we are currently developing a semi-automatic framework for generating vagueness descriptions with VO without having to know its implementation details. In another direction, we plan to evolve VO by looking at its potential links with fuzzy ontologies, identifying more sophisticated vagueness distinctions and phenomena and enabling a higher level of automated reasoning.

## Acknowledgments

The research has been funded from the People Programme (Marie Curie Actions) of the European Union’s 7th Framework Programme P7/2007-2013 under REA grant agreement *n*<sup>o</sup> 286348. We also want to thank all the people who helped us with the evaluation of Vagueness Ontology.

## References

1. P. Alexopoulos and J. Pavlopoulos. A Vague Sense Classifier for Detecting Vague Definitions in Ontologies. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers*, pages 33–37, Gothenburg, Sweden, April 2014. Association for Computational Linguistics.
2. P. Alexopoulos, B. Villazon-Terrazas, and J. Pan. Towards Vagueness-Aware Semantic Data. In F. Bobillo, R. N. Carvalho, P. C. G. da Costa, C. d’Amato, N. Fanizzi, K. B. Laskey, K. J. Laskey, T. Lukasiewicz, T. Martin, M. Nickles, and M. Pool, editors, *URSW*, volume 1073 of *CEUR Workshop Proceedings*, pages 40–45. CEUR-WS.org, 2013.

3. J. Bao, J. Tao, D. L. McGuinness, and P. Smart. Context Representation for the Semantic Web. In *Web Science Conference*, April 2010. Event Dates: 26th April - 27th April 2010.
4. G. Barabucci, A. Di Iorio, S. Peroni, F. Poggi, and F. Vitali. Annotations with EARMARK in practice: A fairy tale. In F. Tomasi and F. Vitali, editors, *Proceedings of the 2013 Workshop on Collaborative Annotations in Shared Environments: metadata, vocabularies and techniques in the Digital Humanities (DH-CASE 2013)*. ACM Press, 2013.
5. M. Benerecetti, P. Bouquet, and C. Ghidini. Contextual Reasoning Distilled. *Journal of Experimental & Theoretical Artificial Intelligence*, 12(3):279–305, July 2000.
6. F. Bobillo and U. Straccia. Fuzzy Ontology Representation using OWL 2. *International Journal of Approximate Reasoning*, 52(7):1073–1094, Oct. 2011.
7. J. Brooke. SUS: A Quick and Dirty Usability Scale. In *Usability Evaluation in Industry*, pages 189–194. Taylor and Francis, London, UK, 1996.
8. P. Buitelaar, M. Sintek, and M. Kiesel. A Multilingual/Multimedia Lexicon Model for Ontologies. In *The Semantic Web: Research and Applications*, pages 502–513, 2006.
9. N. Casellas. Ontology Evaluation through Usability Measures. In R. Meersman, P. Herrero, and T. Dillon, editors, *Proceedings of the On The Move 2009 Workshops*, volume 5872 of *Lecture Notes in Computer Science*, pages 594–603, Berlin, Germany, 2009. Springer.
10. B. Chandrasekaran, J. Josephson, and R. Benjamins. What Are Ontologies and Why Do We Need Them? *IEEE Intelligent Systems*, 14(1):Page 20–26, January - February 1999.
11. R. Falco, A. Gangemi, S. Peroni, and F. Vitali. Modeling OWL ontologies with Graffoo. In *ESWC 2014 Satellite Events - Revised Selected Papers*, Lecture Notes in Computer Science, Berlin, Germany, 2014. Springer.
12. A. Gangemi, J. Lehmann, V. Presutti, M. Nissim, and C. Catenacci. C-ODO: An OWL Meta-Model for Collaborative Ontology Design. In H. Alani, N. Noy, G. Stumme, P. Mika, Y. Sure, and D. Vrandečić, editors, *Workshop on Social and Collaborative Construction of Structured Knowledge (CKC 2007) at WWW 2007*, Banff, Canada, 2007.
13. J. Hartmann, Y. Sure, P. Haase, R. Palma, and M. del Carmen Suárez-Figueroa. OMV – Ontology Metadata Vocabulary. In C. Welty, editor, *Ontology Patterns for the Semantic Web Workshop*, Galway, Ireland, 2005.
14. S. Hellmann, J. Lehmann, S. Auer, and M. Brummer. Integrating NLP Using Linked Data. In H. Alani, L. Kagal, A. Fokoue, P. Groth, C. Biemann, J. X. Parreira, L. Aroyo, N. Noy, C. Welty, and K. Janowicz, editors, *Proceedings of the 12th International Semantic Web Conference (ISWC 2013)*, volume 8219 of *Lecture Notes in Computer Science*, pages 98–113, Berlin, Heidelberg, Germany, 2013. Springer.
15. M. Horridge and P. F. Patel-Schneider. OWL 2 Web Ontology Language: Manchester Syntax (Second Edition). W3C working group note, World Wide Web Consortium, Dec. 2012.
16. D. Hyde. *Vagueness, Logic and Ontology*. Ashgate New Critical Thinking in Philosophy, 2008.
17. J. Kahan and M.-R. Koivunen. Annotea: An Open RDF Infrastructure for Shared Web Annotations. In *Proceedings of the 10th international conference on World Wide Web (WWW 2001)*, pages 623–632, New York, New York, US, 2001. ACM Press.

18. K. Kotis and A. Vouros. Human-centered ontology engineering: The HCOME methodology. *Knowledge and Information Systems*, 10(1):109–131, 2006.
19. T. Lebo, S. Sahoo, and D. McGuinness. PROV-O: The PROV Ontology. W3C recommendation, World Wide Web Consortium, Apr. 2013.
20. J. R. Lewis and J. Sauro. The Factor Structure of the System Usability Scale. In M. Kurosu, editor, *Proceedings of the 1st International Conference on Human Centered Design (HCD 2009)*, volume 5619 of *Lecture Notes in Computer Science*, pages 94–103, Berlin, Heidelberg, Germany, 2009. Springer.
21. T. Lukasiewicz and U. Straccia. Managing Uncertainty and Vagueness in Description Logics for the Semantic Web. *Journal of Web Semantics*, 6(4):291–308, Nov. 2008.
22. E. Montiel-Ponsoda, G. A. de Cea, M. Suarez-Figueroa, R. Palma, W. Peters, and A. Gomez-Perez. LexOMV: An OMV Extension to Capture Multilinguality. In *n Proceedings of the OntoLex07*, pages pp. 118–127, 2007-06.
23. N. Noy, A. Chugh, W. Liu, and M. E. Musen. A Framework for Ontology Evolution in Collaborative Environments. *The Semantic Web - ISWC 2006*, pages 544–558, 2006.
24. R. Palma, P. Haase, O. Corcho, and A. Gómez-Pérez. Change Representation for OWL 2 Ontologies. In R. Hoekstra and P. F. Patel-Schneider, editors, *OWLED*, volume 529 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2008.
25. J. Z. Pan, G. Stamou, G. Stoilos, S. Taylor, and E. Thomas. Scalable Querying Services over Fuzzy Ontologies. In *the Proc. of the 17th International World Wide Web Conference (WWW2008)*, 2008.
26. M. Poveda-Villalon, M. del Carmen Suarez-Figueroa, and A. Gomez-Perez. Validating Ontologies with OOPS! In A. ten Teije, J. Vlker, S. Handschuh, H. Stuckenschmidt, M. d’Aquin, A. Nikolov, N. Aussenac-Gilles, and N. Hernandez, editors, *EKAU*, volume 7603 of *Lecture Notes in Computer Science*, pages 267–281. Springer, 2012.
27. E. Prud’hommeaux and G. Carothers. Turtle - Terse RDF Triple Language. W3C candidate recommendation, World Wide Web Consortium, Feb. 2013.
28. R. Sanderson, P. Ciccarese, and H. Van de Sompel. Designing the W3C Open Annotation Data Model. In *Proceedings of the 5th Annual ACM Web Science Conference (WebSci13)*, pages 366–375. ACM Press, 2013.
29. J. Sauro. *A Practical Guide to the System Usability Scale: Background, Benchmarks & Best Practices*. CreateSpace, Denver, Colorado, USA, 2011.
30. S. Shapiro. *Vagueness in Context*. Oxford University Press, 2006.
31. G. Stoilos, G. Stamou, J. Z. Pan, V. Tzouvaras, and I. Horrocks. Reasoning with Very Expressive Fuzzy Description Logics. *Journal of Artificial Intelligence Research*, 2007.
32. A. Strauss and J. Corbin. *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*. Sage Publications, London, UK, 2nd edition, 1998.
33. C. Thomas and A. Sheth. On the expressiveness of the languages for the semantic web - making a case for a ‘little more’. In E. Sanchez, editor, *Fuzzy Logic and the Semantic Web*, volume 1 of *Capturing Intelligence*, pages 3 – 20. Elsevier, 2006.
34. D. Vrandečić, H. S. Pinto, Y. Sure, and C. Tempich. The DILIGENT Knowledge Processes. *Journal of Knowledge Management*, 9(5):85–96, Oct. 2005.
35. L. A. Zadeh. From Search Engines to Question-Answering Systems - The Need for New Tools. In E. M. Ruiz, J. Segovia, and P. S. Szczepaniak, editors, *AWIC*, volume 2663 of *Lecture Notes in Computer Science*, pages 15–17. Springer, 2003.