

## Browsing Through OWL Domain Ontologies

Christian Kop

Applied Informatics

Alpen-Adria-Universitaet Klagenfurt

Klagenfurt, Austria

chris@ifit.uni-klu.ac.at

**Abstract**— There is a trend to see ontologies not as one big specification of a certain domain but to see it as a network of interrelated units. This paper will propose an approach and a tool, which allows the user to navigate through a network of ontology documents. The user will be supported by two different presentation techniques (graphical visualization and verbalization). Furthermore, s/he will be provided with information about relevant (key) classes and with a textual summary report of the ontology. Apart from the visualization of specific OWL domain ontologies, s/he will also get support for the visualization of the ontology network in which a domain ontology specification is just a unit, which is related to other units. Hence, a tool is introduced, which allows to present ontologies in different ways.

**Keywords**- *semantic web; OWL; verbalization; summary; relevant classes; key classes; ontologies; visualization.*

### I. INTRODUCTION

With the Semantic Web Initiative [14], the Internet more and more becomes an Internet, which contains ontologies. Furthermore,, ontologies do not appear solely but are connected to each other. Such ontologies were mainly generated to be machine interpretable. But also a human reader must be able to browse through this network of ontologies. The tool, which is described here, does not provide mechanism for automatic reasoning but techniques to present the ontology to human readers. Especially within a network of ontologies it is necessary to give an overview of the ontology network and to provide a combination of visualization techniques for a node (ontology document) within this network. Therefore, one of the first changes to traditional tools is the visualization of an overview of the document network itself before visualizing the ontology inside a certain document. Furthermore,, it must be guaranteed that a user gets information, which node (document) s/he has successfully visited and which document is currently opened. To summarize, in this paper the following presentation strategies are proposed:

- Graphical representation of the import relationships between ontology documents

- Graphical representation of the user behavior (navigation visualization) within the document network.
- Graphical representation of the ontology specification for a specific ontology document.
- Verbalization of this specification
- A list of relevant classes
- Textual summary of the formal ontology specification.

Whereas other tools focus especially on either visualizing or verbalizing an OWL ontology itself, this approach focuses on the combination of these techniques. Furthermore,, it provides the possibility to show an OWL ontology as a network of interrelated OWL documents if the ontology imports concepts from other OWL documents. Therefore, it is possible to navigate through the interrelated OWL documents.

The ontology specification itself is presented on two additional levels. It is presented graphically or verbalized as a whole. If a user wants to see details on an OWL class then they are shown graphically and they are verbalized.

In order to describe such a tool and approach, the paper is structured as follows. In the next section, an overview of related work is given. Section III continues with the description of the document network visualization. Section IV starts with the first detailed view for a specific document node. This view is a graphical one. In Section V, a specific view is introduced, which lists all OWL classes together with the possibility to provide the user an understanding of the relevant classes of the ontology. This is based on measures, which are directly derived from the ontology structure. In this section also traditional views are introduced. The other important presentation strategy (verbalization) will be presented in detail in Section VI. Section VII combines the verbalization strategy and measures for relevant classes to introduce a view of a textual summary for an ontology. Section VIII gives a final discussion of the features of the presented tool. Section IX summarizes this approach and gives an outlook to future work.

## II. RELATED WORK

Since the tool covers different techniques (e.g., graphical visualization, verbalization), the related work section is divided into subsections, which cover these aspects.

### A. Visualization of Ontologies and Networked Ontologies

Concerning the research field of the visualization of an ontology specification, a lot of work has been done so far. Lanzenberger, Sampson and Rester summarized this work [16]. Beside traditional two-dimensional graphical views there is a trend to introduce 3D visualization techniques for large ontologies. The tools OntoViz [17], OWLViz [18] and Ontobroker [21] visualize ontologies with two-dimensional graphs. Tools like Ontorama [19] and OntoSphere 3D [20] provide three-dimensional representation techniques. Furthermore, visualization is not only used to present the ontology content but is also used to view specific aspects of the content. For instance the tool AlViz introduced by Lanzenberger et al. in [22] and [23] visualizes the alignment of two ontologies. The approaches described by Falconer et al. [24] and Gilson et al. [25] visualize the mapping of ontologies. Garcia et al. proposes how the coupling of ontologies can be visualized [26].

Ontologies are nowadays not seen as one big specification of a certain domain but they are seen as a network of interrelated units. The NEON Approach ([www.neon-project.org](http://www.neon-project.org)) is a research project, which motivates networked ontologies. Here it is even proposed that ontologies should be splitted into modules [27]. Currently however, an ontology document itself is still the most used container, which can be treated as a bigger unit. The approach described in this paper therefore follows this traditional view of modularization. OWL classes and object properties, which belong together are specified in the same document. If classes and object properties are needed from another OWL document, then this external document is imported. In this sense, an ontology documents network is established. In order to browse such a network, browsing must start at the level of OWL documents.

### B. Verbalization

Strategies to verbalize ontologies are described in Fuchs et al. [5] and Hewlett et al. [6]. These approaches mainly focus on an optimal natural language verbalization of OWL constructs itself like *subClassOf*, *intersectionOf*, *unionOf*, cardinality restrictions etc. Although, this helps to read the whole ontology it still looks artificial since the labels of OWL classes and OWL object properties are not transcribed. In their work on verbalization of OWL 1.1., Karljurand and Fuchs still conclude that object property and class labels hopefully will become more English-like over time. Luckily an analysis of many OWL ontologies made by Mellish and Sun showed that in many cases nouns for classes and verbs for object properties are used [11]. Although object property labels contain a verb many name variations can exist. Therefore, in [4] linguistically

motivated guidelines and naming conventions were proposed, which must be used for object property labels and class labels. Information of individuals is verbalized in the research work of Bontcheva [3].

### C. Relevant Class Measures

Interesting related work for this case was done by Bezerra et al. [2] and Huang et al. [7]. They introduced key classes, which are similar to relevant classes in order to estimate the quality of an ontology. The weighted relationships of a class are counted in [7]. The weight is determined from the relationships that can be inherited from the super classes. Then these weights are forwarded to the involved classes of these relationships. In [2], the number of direct children is counted. Vrana and Mach propose to return a vector of terms instead of the whole ontology [30]. This vector can be seen as a summary, which is based on the input of a user who searches for keywords in ontologies. More detailed research on key concepts are provided by Zhang et al. [31] and Peroni et al. [29]. Zhang et al. summarizes RDF ontologies. It returns a graph of salient RDF sentences. The approach proposed by Peroni et al. integrates topological measures (density and coverage of concepts) with statistical measures (popularity of a concept) as well as cognitive criteria (natural categories). Topological measures rely on the structure of the ontology. The included cognitive criteria is based on the idea that simple single words represent more likely key concepts. However it turned out, that this criteria does not work so well. The popularity measure is estimated by counting the hits of several ontology concepts in Yahoo. Then the hits are compared. An ontology concept, which has more hits than another is seen as more popular. This measure was introduced to optimize the output with regard to human experts. The authors in [29] found out, that human experts tend to prefer more common knowledge terms rather than special terms. In cases where common knowledge terms and special terms are equal candidates for key concepts, they wanted to ensure that the common knowledge term is preferred in the selection process by the algorithm. Centered concepts play an important role for estimating clusters in conceptual models (see the work of Moody et al.) [12] [13]. Since conceptual models depend on relationships between entity types (classes), for each entity type its relationship is counted.

### D. Differences to Related Work

Although there is a lot of work, which focuses on graphical representation of ontologies, a visualization of interrelated ontology documents was not found in literature. This approach therefore not only presents a visualization of the ontology itself but also focuses on the visualization of the import relationships between OWL documents. In fact, this is the starting point of the browser. In this feature, the user can enter the link to an ontology document. Then s/he is provided with the document itself and all the other OWL

documents, which are needed for “imports” in the chosen OWL document. From this first view s/he can start to navigate and explore the OWL document network. A more detailed description of how this can be done, will be presented in Section III. Of course s/he can let the tool present the content of each document. This is done graphically or by using verbalization. Actually, a two-dimensional graphical representation is provided for the graphical representation of the ontology content.

The verbalization strategy proposed here is a refinement of [4]. It allows some more labeling freedom, though some linguistic guidelines are still needed to achieve good verbalization results. Finally this approach differs from the approach described in [3] since this approach verbalizes OWL classes and not individuals.

The determination of relevant classes in this approach relies on the ontology structure itself. A class is only a relevant (key) class if this can be automatically determined from the content of the ontology itself. A popularity measure is not used in this approach since it was the aim that only the structure and no external resource should be used. Whereas in [29] only the best N key concepts are returned, this approach does not select only N concepts but categorizes concepts into relevant (= key) concepts and considerable concepts. Hence, beside an inner circle of concepts it also has an outer circle of considerable concepts, which could be candidates too. Furthermore, in one view of the tool it is also possible to order all the concepts according to its relevance. In this view nothing gets lost. As an alternative to the number of children [2], the number of all successors is calculated. This alternative achieves that the relevance is treated globally since the whole sub tree and not only the children are considered. The calculation of the weight for relationships as described in [7] is not taken as the only measure, since it only works if classes have object properties. However, OWL classes do not necessarily need such object property relationships. In fact, there are also ontologies, which only consist of a class/subclass taxonomy. If object properties are specified in an ontology, then it will be described that simpler measures are sufficient. The paper introduces several other statistical measures for classes (e.g., number of instances, number of restriction and disjoints). Instead of the RDF basis [31] it focuses on specific OWL features.

All verbalization strategies today verbalize the whole ontology content. On one hand it is an advantage since the human reader does not have to understand the formal and artificial constructs of the ontology. On the other hand however, it is still necessary to read the whole ontology. To use a metaphor, this can be compared with presenting the reader a full “news paper article”, though at the beginning, it might be better to present him only an abstract or the headline (textual summary).

Regarding textual summary generation, it can be said, no approach is known, which currently generates a textual summary out of the ontology structure. According to an

evaluation made in [1] a combination of two measures out of the measures for relevant class determination were used as a basis for textual summary generation.

Finally, in this paper it is proposed to combine all the above mentioned representation techniques together in one tool.

### III. MAP OF INTERRELATED OWL DOCUMENTS

Each OWL document is related with other OWL documents from which it imports necessary other specifications. For instance, the wine ontology imports PotableLiquid from the food ontology. This resource is needed to specify that a Wine is a subclass of PotableLiquid. Another example: The climaticzone ontology at the designpatterns web site needs the ontologies aquatic resource, cpannotationschema and observation. The tool visualizes this information.

#### A. Visualization of the Document Network

Viewing such a network is done, using the header of the OWL specification in which the imports are specified. Instead of presenting the whole ontology it firstly scans the document for the section with the import specifications.

It then paints the document itself as a node (rectangle). If the document imports concepts from other documents then these import relationships are visualized as directed edges pointing from the document, which needs the import to the document, which provides concepts for the import. Furthermore, the system does not show the user the whole network at once. Instead it starts with the chosen OWL document and its direct neighbors (i.e., those documents, which are needed for imports). If the user wants to see more of the network, s/he has to expand those documents, in which s/he is interested in. This stepwise visualization document network has the following reason: It cannot be trusted that all the nodes in the network are available since the nodes (OWL documents) might be spread over different servers on the Web. Some servers might be not available. Therefore the system firstly only provides the start node and the direct neighbors of the start node (document). This is possible since the start node is available and the specification of the imports is given inside the start document. Figure 1 shows the situation at the beginning for the “Climatic Zone” ontology, which was taken from the ontology pattern web side (see: [32]).

#### B. Operations Provided in the Document Network

If the user selects one document s/he can open a popup menu with the right mouse button. Then s/he gets menu options for operations, which can be applied on the selected document. Namely, it is possible to

- Open a detailed graphical view containing the ontology (for more details see Section IV)
- Open a list view where all the ontology elements are listed (see Section V) and verbalized (Section VI).

- Open a summary view where a natural language summary is generated (see Section VII)
- Get all the import details for all the needed imports of the document. This operation is also available for the import edges between the OWL documents.
- Expand a node. This operation supports the navigation through the network (see Figure 2).

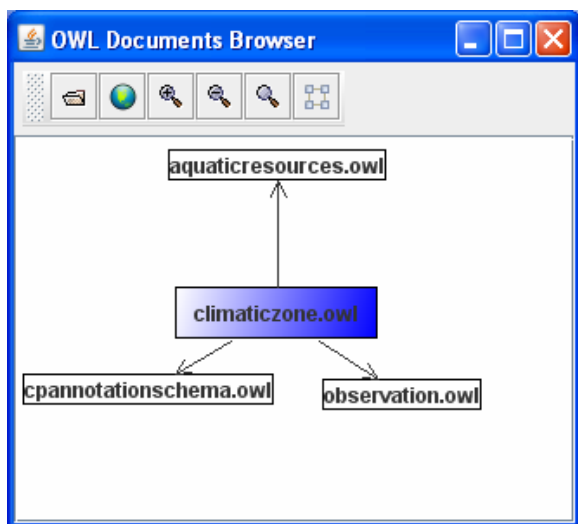


Figure 1. Relationships between OWL documents

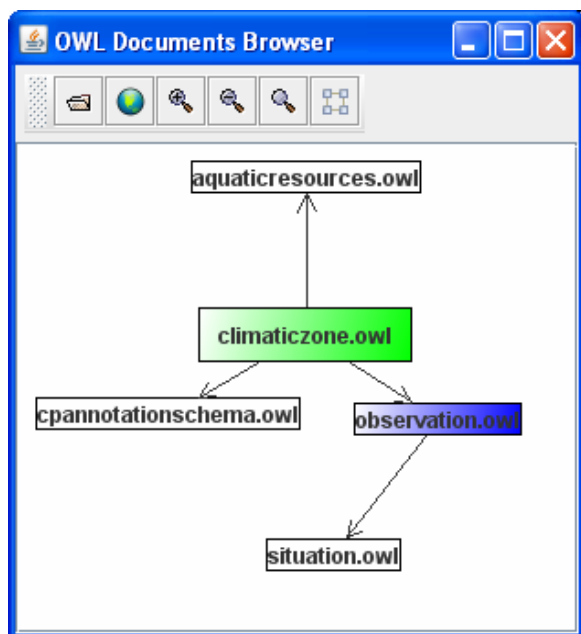


Figure 2. Expansion of observation.owl

### C. Representation of Imports

The representation of imports has the same structure for both the OWL documents and the import edges. In both cases, the user gets a list of OWL elements and the resources (imported documents), from which they were imported. Only the resulting list is different. If the user chooses this list for an OWL document, then a list of elements from all the surrounding neighbors of that OWL document is provided (these neighbors are the documents, from which elements are imported). If s/he selects an edge then s/he only gets the list of elements from the document to which the edge points (the imported document). Figure 3 gives an example for the edge pointing from climaticzone.owl to observation.owl.

Imported Elements		Verbalization of Imports		
Type	Name	Relationship	Resource	Imported from
Class	AquaticResource	owl:disjointWith	Parameter	<a href="http://www.onto...">http://www.onto...</a>
Class	AquaticResource	owl:disjointWith	Observation	<a href="http://www.onto...">http://www.onto...</a>
ObjectProperty	hasClimaticZone	rdfs:subPropertyOf	hasParameter	<a href="http://www.onto...">http://www.onto...</a>
ObjectProperty	isClimaticZoneOf	rdfs:subPropertyOf	isParameterOf	<a href="http://www.onto...">http://www.onto...</a>
ObjectProperty	isResourceOf	rdfs:subPropertyOf	hasObservation	<a href="http://www.onto...">http://www.onto...</a>
ObjectProperty	hasResource	rdfs:subPropertyOf	isObservationOf	<a href="http://www.onto...">http://www.onto...</a>

Figure 3. Visualization of imports

The first column specifies the type of the OWL element (e.g., is it an OWL class or an object property), which needs a resource from another OWL document. The second column gives the name of the element. The third column names the kind of “relationship” to the imported element. The column resource gives the name of the imported element. The last column specifies the resource locator.

There was also an attempt here to verbalize this kind of information (second tab in the screenshot of Figure 3).

### D. Visualization of the Navigation

Beside the visualization of the document network structure, also the behavior of the user is visualized. First of all, the document with which the user starts appears larger than the other documents (see also Figure 1).

For traditional web sites it is already an established strategy that links, which have not been followed yet appear differently to links, which were already opened by the user. Since this strategy has proven to be very successful, it was adopted for the navigation visualization. A coloring system was introduced to show

- Where the user currently is and executes an operation (e.g., the node s/he opens to browse through the OWL specification),
- Which nodes has been opened successfully in the past and
- Which nodes (documents) are not available (i.e., the server was down and hence it was not possible to open this specific ontology).

If a node is opened in order to browse the ontology specification itself or if it is only expanded, then the node is colored “blue”.

If the node was opened successfully in the past, then this node remains colored “green”, but it can change to blue if the user once again opens exactly this node. In this case also the previously opened blue node changes his color to green.

If s/he is not able to open or expand a node then the node is colored “red”. Such a situation appears if the server, on which the ontology document is stored, is not available.

#### IV. DETAILED GRAPHICAL VIEW

The graphical view itself is divided into four separate views: A view of the subclass taxonomy, a view of the object properties, a view for simple restrictions based on allValuesFrom, someValuesFrom and hasValue as well as a view for disjoints. The first two views were made since taxonomies play an important role in OWL specifications. Therefore the subclass relationships are visualized in an extra view and are not mixed up with the object properties. The view for disjoints and the view for the three types of restrictions were also separated for the same reason. In all the graphical views each OWL class is drawn as a rectangular node in the graph.

##### A. Class Taxonomy View

The graph taxonomy view only consists of classes and their specializations. In the view two types of specializations are distinguished and therefore are presented differently. For explicit generalization hierarchies OWL uses “subClassOf”. In the graphical view, this kind of relationship is drawn as an edge between the OWL class nodes with a solid line and a solid triangle, which points from the specialization to the general OWL class.

Beside this, it is also possible in OWL ontologies that generalization hierarchies are stated implicitly. Implicit specification can be expressed in terms of an equivalent class and an intersection. The next OWL specification part is an example for such a situation. “CheesyPizza” is indirectly specified to be equivalent to an intersection between Pizza itself and a restriction on the object property “hasTopping”.

```
<owl:Class rdf:ID="CheesyPizza">
  <owl:equivalentClass>
    <owl:Class>
      <owl:intersectionOf
        rdf:parseType="Collection">
          <owl:Restriction>
            <owl:someValuesFrom
              rdf:resource="#CheeseTopping"/>
            <owl:onProperty>
              <owl:ObjectProperty
                rdf:about="#hasTopping"/>
              </owl:onProperty>
            </owl:Restriction>
          <owl:Class rdf:about="#Pizza"/>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>
```

```
...
  </owl:intersectionOf>
</owl:Class>
</owl:equivalentClass>
</owl:Class>
```

Such an indirect specialization is drawn as an edge with a dashed (red colored) line and a non solid triangle pointing from the specialization to the generalization. Figure 4 shows parts of the graphical taxonomy view for the pizza ontology.

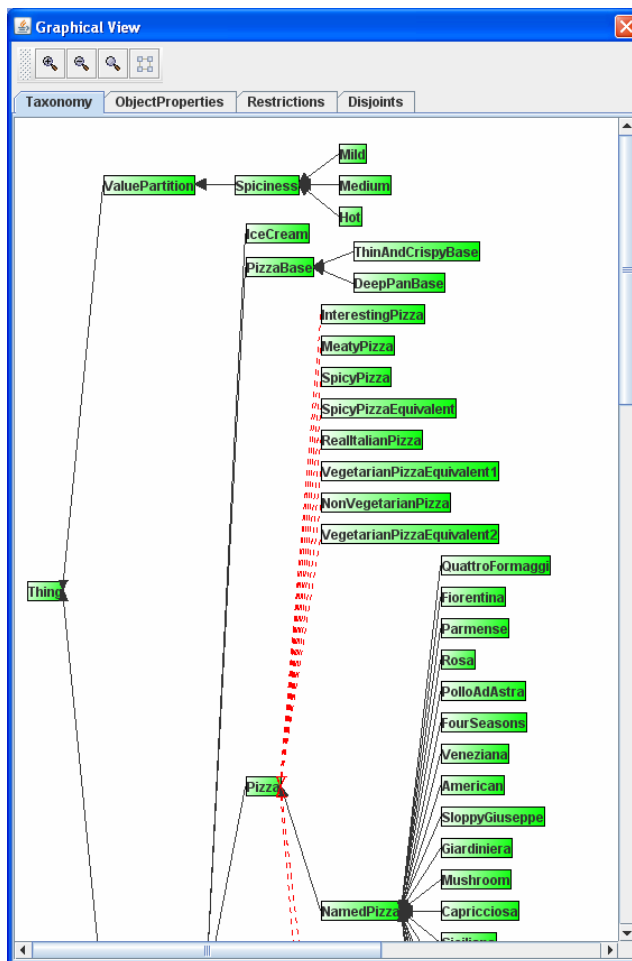


Figure 4. OWL class taxonomy view

##### B. Object Property View

As specified with the name, the object property view shows the OWL object properties. The fact that object properties contain a domain and a range is presented with a directed edge. The arrow of the edge points from the OWL domain to the OWL range class. Figure 5 shows such a situation.

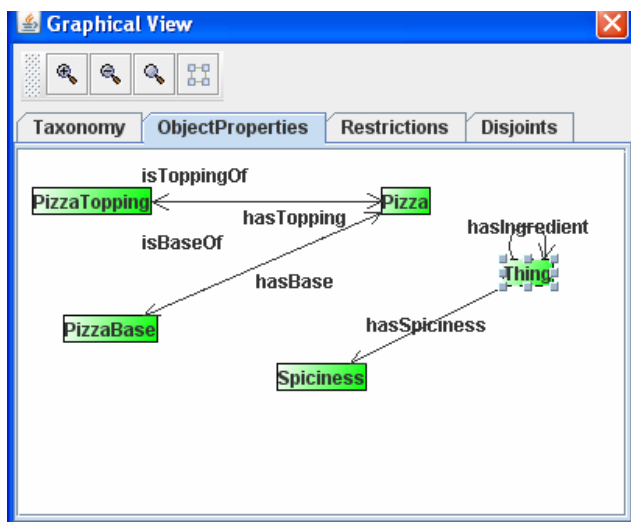


Figure 5. Object property view

C. Restrictions and Disjoints View

The last two tabs contain views on restrictions (namely: someValuesFrom, allValuesFrom, hasValue) and disjoints. The restrictions as well as the disjoints are presented as dashed lines between the concepts. Furthermore, restrictions can be distinguished by different colors and a different edge ending at the position of the class for which the restriction was defined. A solid circle represents an allValuesFrom restriction. A non solid circle represents a someValuesFrom restriction and no circle is used for a hasValue restriction. (see Figure 6 for graphical visualization of allValuesFrom and someValuesFrom restrictions).

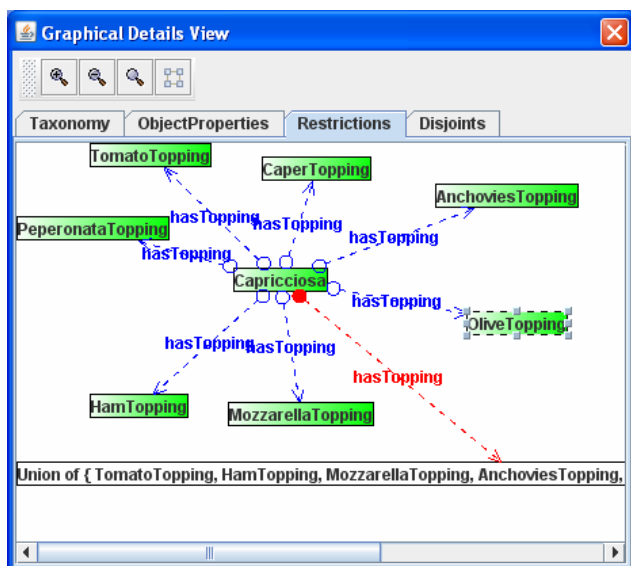


Figure 6. Restrictions view for OWL class Capricciosa

D. OWL Class Graphical Details View

The graphical details view for one OWL class has the same representation features as the graphical view for all OWL elements. The difference is, that the representation of the taxonomy, object property, restrictions and disjoints is shown for one selected OWL class. Figure 6 shows the restrictions for the selected OWL class “Capricciosa” in the pizza ontology.

V. RELEVANT CLASSES AND OTHER TABULAR REPRESENTATIONS

Beside the graphical views, the tool provides the human reader also with the following tabular and verbalized views: the OWL classes together with their relevance, a class taxonomy tree view, instances, restrictions and disjoints.

A. Relevant Classes

Several kinds of measures to sort OWL classes for their relevance are introduced in this paper.

- Weighted number of successors (wNS),
- Number of object properties (P),
- Weighted number of object properties (wP),
- Number of children (NC),
- Number of restrictions and disjoints (R/D),
- Instances of an OWL class (I),
- Total value (T).

The weighted number of successors (wNS) focuses on the subclass hierarchy. The successors (NS) of a class are counted and weighted. This avoids that OWL classes at the top of the hierarchy will always be the winners.

For the weighting factor, the distance of a certain class from the root Thing is taken (= dfr) and it is divided by the maximum distance to a leaf (= maxdtl). The distances are calculated by counting the edges from the root to the leafs respectively. The weighted number of successors for a class X hence is calculated as:

$$wNS(X) = NS * (dfr / maxdtl) \tag{1}$$

In the pizza ontology the distance from root to pizza is 2 (i.e., Thing → DomainConcept → Pizza). The maximum distance to a leaf is also 2. The weighting factor therefore is 2 / 2 = 1. Pizza itself has 34 successors. Hence, wNS(pizza) is 34.

The number of object properties for a domain class (P) considers the object property relationships from a domain class to its range class. It is assumed that the domain class is more likely a relevant class than the range class in the object property. Therefore each object property is counted only for domain classes. For example in the pizza ontology the class Pizza gets the value 2 for P since pizza is the domain class in two object properties (pizza has pizza topping and pizza has pizza base).

The weighted number of object properties (wP) is a refined version of the measure P. Instead of incrementing the counter by 1 for each class, which is involved as a

domain in an object property, the counter is incremented by the wNS value of the range. However, if the value is 0 then wP is incremented by 1 and it degrades to P. With this strategy the importance (weight) of the range is also forwarded to the domain.

Instances of an OWL class (I) are a third strategy to estimate the relevance of a class. In the pizza ontology the class *Country* has 5 instances (I). The number of children was taken from [2] and just counts the direct children of a class.

The number of restrictions and disjoints are calculated by incrementing a counter for an OWL class whenever a restriction (someValuesFrom, allValuesFrom, hasValue) or disjoint is specified for that class.

The default value of T is a combination of the wNS and wP measure. It is not a simple count but it divides the counted result into an ordinal value system ranging from 1 to 3. A more detailed description of the determination of these values will be given in Section VII. The value for T can also be set to be the sum of any selection of the measures wNS, NC, wP, P, R/D, I. Figure 7 shows how relevant classes are visualized in a listing. The elements in the listing can be sorted according to each of the measures.

#### B. Other Visualizations

Beside the visualization of relevant classes, the user can also choose among the following other tabular and J-Tree-based visualizations (object properties, class tree, class details, restrictions, disjoints and instances).

The tabular object property representation lists object properties. The table has three columns: object property column, domain and range column. The object properties together with their domain and range are verbalized. A J-Tree based representation is provided for the class taxonomy and the details for a class. In the J-Tree for the class taxonomy, each class and their subclasses are visualized as folders and subfolders. If a class has instances, also the instances are presented as leaves in the J-Tree. In the J-Tree for the details of all classes, the classes once again appear as folders. If such a class folder is opened, the sub folders contain the restrictions, instances, object properties sub classes etc., for the selected class.

The restriction are listed also in a tabular representation. This listing has three columns. One column is for the class on which the restriction is defined. The other column is for the restriction category and the third column for the specification of the restriction. The content of this last column is once again verbalized.

The instance list is another tabular representation, which focuses on the instances. It has two columns. The first column contains the instances. The second column contains the classes to which the instances belong.

In each of the tabular representations it is possible to sort by each column individually. For instance, object property specifications can be sorted either by their domain or range.

Class Name	wP	NC	P	I	R/D	T
Named Pizza	69	0	23	0	0	3
Pizza	34	29	11	2	0	4
Pizza Topping	25	34	10	1	0	3
Vegetable Topping	22	0	14	0	0	7
Cheese Topping	18	0	6	0	0	7
Domain Concept	18	0	5	0	0	1
Pepper Topping	10	0	4	0	0	12
Fish Topping	9	0	3	0	0	8
Tomato Topping	8	0	2	0	0	13
Meat Topping	7	0	4	0	0	7
Herb Spice Topping	6	0	2	0	0	7
Spiciness	6	0	3	0	0	0
Green Pepper Topping	5	0	1	0	0	3
Ham Topping	4	0	1	0	0	3
Onion Topping	4	0	1	0	0	13

Figure 7. Relevant classes list

#### C. OWL Class Details

For presenting details of an OWL class, most of the views in Section B can be reused. Only the relevant classes view and the taxonomy view make no sense in this context. The views on restrictions, disjoints, class details, instances and object properties can be applied for the selected class since only information is listed, which belongs to that class.

### VI. VERBALIZATION OF AN ONTOLOGY

In the previous section, the listings presented also verbalized information. Therefore, this section explains the details.

#### A. Problems of Verbalization

In the verbalization step the following two problems are well known and discussed in literature [4],[5],[7],[9],[11]:

1. How can the formal knowledge representation be transformed into a natural language representation?
2. How can OWL class and object property labels be transformed into a natural language representation?

The first problem addresses the different OWL elements and constructs to define formal expressions (e.g., *Intersection*, *Union*, *someValuesFrom*, *allValuesFrom* etc.). Previous research work already provided solutions for that problem.

The second problem occurs since the human OWL designers cannot be influenced to use proper and standardized labels for classes and object properties. Many variations exist. In [4], the verbalization strategy recommends that classes and object properties are labeled exactly in a specific format. Namely, classes should be

nouns in singular. If there are compound nouns, then they must be separated using an upper character (e.g., *VegetarianPizza*) object properties must start with a verb. If a preposition is needed, then the preposition must be explicitly specified in the label. Once again single words must be separated by an upper character (*workWith*). The language used for all the labels must be English.

#### B. Verbalization of Classes and ObjectProperties,

This approach addresses the second problem and refines the work in [4] and [6]. In order to produce better results and give ontology designers more freedom of defining the labels, the following must be done:

1. Find the separation strategy and separate accordingly.
2. Remove redundant information in the object property labels.
3. Add additional information to the object property.
4. Add the article “a” or “an” respectively to the two involved classes of a relationship.

Most ontology engineers prefer to separate words by using upper case characters (e.g., *VegetarianPizza*, *worksWith* etc.). Some also use separators like “\_” or “-” (e.g., *vegetarian\_pizza*). If the separator strategy is detected, the words can be separated to their natural language form (e.g., “*vegetarian pizza*“, “*works with*” etc.).

If ontology engineers only would use verbs in object property labels the verbalization would be much easier. Unfortunately, they often add information about the domain and range to the object property itself. This is often necessary to avoid different object properties with the same labels (i.e., several object properties with the label “*has*”) but it produces artificial results in those verbalization approaches, which do not handle this. Particularly if the range (domain) or closing parts of the range (domain) are also mentioned in the object property label, then this must be removed (e.g., “*has*” instead of “*hasTopping*”). The verbalization result is much better if such changes on the object property labels are made (e.g., “*Pizza has Pizza Topping*” instead of “*Pizza has topping Pizza Topping*”).

Additional information is necessary if there is an object property, which contains the verb *has* together with a noun but the noun is not part of the range (e.g., *person has father man*). This is verbalized to *person has a father, which is a man*. However, this strategy fails, if the range class is not a noun. In some of the ontologies adjectives were specified as range concepts in object properties (e.g., *Wine has color Green*). In this case, only an additional lexicon with adjectives can help to avoid a wrong verbalization. For instance this can be solved as proposed by Sugumaran et al. [28]. There WordNet was taken as the lexicon. A word like “*mild*” can be searched in the lexicon. The lexicon returns all the meanings of “*mild*” together with the word category of this meaning and a measure how often the meaning is used. If the word category of the word with the most used

meaning is an adjective, then this can be considered for verbalization.

Finally, if the domain and the range of the object property are common nouns then, to each of the two classes involved, the article “a” or “an” respectively is added. If an adjective or a mass noun was used to label an OWL class, then no article must be added.

With these strategies of verbalizing labels, the ontology designer has much more flexibility to name the object properties and OWL classes. (see Figure 8)

Object Property	Domain	Range
A Pizza Base is base of a Pizza	Pizza Base	Pizza
Anything has a Spiciness	Anything	Spiciness
A Pizza has a Pizza Base	Pizza	Pizza Base
A Pizza Topping is topping of a Pi...	Pizza Topping	Pizza
Anything has an Ingredient which ...	Anything	Anything
Anything is ingredient of Anything	Anything	Anything
Anything has a Country Of Origin ...	Anything	Anything
A Pizza has a Pizza Topping	Pizza	Pizza Topping

Figure 8: object property verbalization

#### C. Verbalization of Restrictions and Disjoints

The verbalization of the restrictions and disjoints reuses the strategies for OWL class and Object Property verbalization. Currently it is based on patterns found in the OWL ontology. The pattern:

```
<owl:Class rdf:ID="Anjou">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        rdf:resource="#hasColor"/>
      <owl:hasValue rdf:resource="#Rose"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

is verbalized to “*Anjou has a Color, which is (has value) Rose*”. Particularly, if within the class description of Anjou a HasValue-Restriction of on the Object Property is found then it is transcribed as shown above.

If a AllValue-Restriction together with a Union operator is found within an OWL class specification as shown below,

```
<owl:Class rdf:about="#AmericanHot">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty>
          rdf:about="#hasTopping"/>
        </owl:onProperty>
```



```

<owl:allValuesFrom>
  <owl:Class>
    <owl:unionOf rdf:parseType="Collection">
      <owl:Class
        rdf:ID="HotGreenPepperTopping"/>
      <owl:Class
        rdf:about="#JalapenoPepperTopping"/>
      <owl:Class
        rdf:about="#MozzarellaTopping"/>
      <owl:Class
        rdf:about="#PeperoniSausageTopping"/>
      <owl:Class
        rdf:about="#TomatoTopping"/>
    </owl:unionOf>
  </owl:Class>
</owl:allValuesFrom>
</owl:Restriction>
</rdfs:subClassOf>
</owl:Class>

```

then it is verbalized to

*"An American Hot has a Topping, which is (all values from) the Union of { Hot Green Pepper Topping, Jalapeno Pepper Topping, Mozzarella Topping, Peperoni Sausage Topping, Tomato Topping }"*

Disjoint specifications like

```

<owl:Class rdf:about="#AmericanHot">
  <owl:disjointWith>
    <owl:Class rdf:about="#Napoletana"/>
  </owl:disjointWith>
</owl:Class>

```

are verbalized as *"An American Hot is disjoint with a Napoletana"*.

#### D. Verbalization of Imports

In Section III it was mentioned that the list of imports is also verbalized. This is done by using sentence templates in which the specific import information (e.g., imported element, element that refers to an imported element etc.) is inserted. The template looks as follows: *The (Class / ObjectProperty) <name> (is disjoint with / is a Subproperty of) is a sub class of / ...) the Resource <name>, which is imported from <import address>.*

### VII. TEXTUAL ONTOLOGY SUMMARY

As pointed out in [1] a textual summary of an ontology is useful if a human reader just wants to get a first impression of the specification. Particularly, it can be more compared with a situation you will find if people read news paper articles. Many people look at the title, headlines or the short abstract before they start reading the article itself. Hence, what is needed is a technique that automatically summarizes the relevant content for the human reader. In other words, such a tool must be able to generate and verbalize a subset of the ontology structure that can be seen as an abstract, summary

or headline of the ontology. Only if the ontology is verbalized and the information about the relevant OWL classes is summarized, then the human reader can derive his first impression quickly and efficiently. A summary or abstract of an OWL ontology can be generated by combining the determination of relevant (key) classes with verbalization strategies. Particularly, the summary generation consists of the following steps:

1. Calculation of class relevance based on measures,
2. Categorization if classes are relevant or not,
3. Verbalization of classes and their object properties.

The Step 1 was already described in Section V. Step 3 was described in more detail in Section VI. Therefore this section concentrates on Step 2. In [1], the combination of the weighted number of successors and the weighted number of object properties were seen as a good basis for textual summary generation.

Therefore, once each class got a measured value for wNS and wP (step 1) they are divided into the three categories.

- Relevant class (must be in the summary)
- Considerable class (can be in the summary)
- Not relevant class (not included in the summary)

This is done for wNS and wP separately. In order to categorize it is necessary to know the maximum measured value for wNS (wP respectively) for a certain class. This maximum is taken as the basis. It represents 100 % of the reachable measured values for wNS (wP). The wNS (wP) values for all other classes are now compared with the maximum (= 100 %). A percentage for these values is calculated. If a class reaches a percentage higher than 65% then it is treated as a relevant class for a summary. If the calculated percentage is between 33 % and 65 % then it is a considerable class. Otherwise it is not relevant.

The class with the maximum wNS in the pizza example is *NamedPizza* (wNS = 69). It is followed by *Pizza* (wNS = 34), *PizzaTopping* (wNS = 25). With these values for wNS, *NamedPizza* is a relevant class. *Pizza* and *PizzaTopping* can be seen as considerable classes since they do not have a wNS value higher than 45 (65 %). The other classes are not relevant at all according to their wNS values.

Applying the wP measure, the classes with the maximum wP value 34 are *PizzaTopping* and *PizzaBase*. Both classes have this value since they are involved in an object property as domains where the range is *Pizza*. *Pizza* follows with a wP value of 29. Here all three classes are relevant classes.

With the two measures (wNS, wP) and the distinction between relevant and considerable classes, it can be parameterized how restrictive the generated summary is. The summary can be generated on the basis of one of the measures only (i.e., either wNS or wP) or based on the combination of the two measures together. The latter can be understood as a union of wNS and wP. Furthermore, it can be decided if only relevant classes will appear in the

summary or relevant and considerable classes will be listed. The next table outlines the several possible alternatives.

TABLE I: ALTERNATIVES FOR SUMMARY REPORT

	relevant only	relevant + considerable
wNS	(1)	(2)
wP	(3)	(4)
wNS + wP	(5)	(6)

In the following the combination of wNS + wP as a basis for listening relevant and considerable classes (6) is described in detail. It contains the other alternative reports (1) – (5) and thus is the most general report.

Hence, if the alternative (6) is taken, then the summary report would not only return verbalizations of relevant classes but also verbalization of considerable classes. That means:

- A class is a relevant class if it can be categorized as a relevant class on the basis of at least one of the measures (wNS, wP).
- A class is a considerable class if it is not a relevant class and it is categorized as a considerable class by at least one of the measures (wNS, wP).

If now the verbalization strategies (Section VI) are reused, then a summary can be generated by an introduction template. “The most relevant classes are <list of relevant classes> followed by <list of considerable classes>”. If only the relevant classes are needed (alternatives (1), (3), (5) in Table VII) then no considerable classes are mentioned in the generated output.

Afterwards each of the listed classes is verbalized. Particularly, its relationship to its more general classes is named (e.g., a Pizza is a Domain Concept). Furthermore,, verbalization of those object properties is given where the relevant (considerable) class is involved explicitly as a domain (e.g., a Pizza has a Pizza Topping). Figure 9 shows a textual summary report for the Pizza ontology.

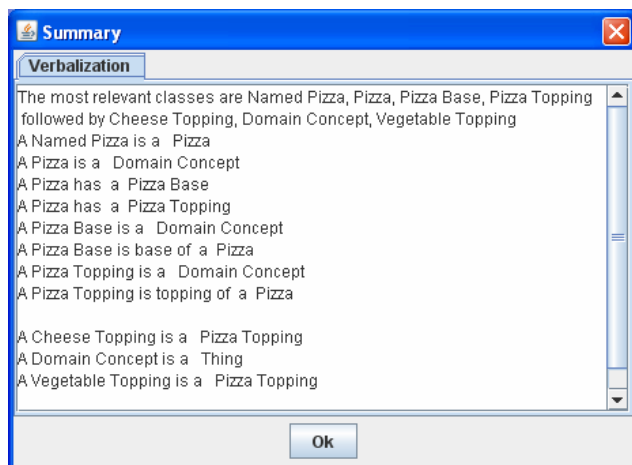


Figure 9. Textual summary

### VIII. DISCUSSION

The ontology documents browser was tested with ontologies on the website [32] and with the ontologies mentioned in [1] and found in [15]. Especially in the case of the ontology design pattern web site, it turned out that it is very helpful if somebody wants to get an overview of the surrounding of a specific ontology (e.g., “climaticzone.owl”). The tool has the advantage to derive this information directly from the content of the chosen ontology. Then it visualizes the information graphically with the chosen ontology in the center and the ontologies needed for import as satellites of this ontology. Furthermore,, the tool provides him with navigation visualization. The starting point (starting document) is always presented larger than other documents. Successfully visited nodes (documents) are colored differently to nodes, which were not visited or nodes, which could not be visited (i.e., web server is down). Thus, the user gets a spatial overview of his navigation. S/he can even browse through the documents without looking inside one document. The users always have the spatial overview of the documents network.

In [1] the calculation of relevant classes was tested. It could be shown, that the calculation of relevant classes reflects the content structure of the ontology. It was therefore also “natural” to use these classes for the textual summary of the ontology content.

Regarding the verbalization, this paper focused on verbalization of labels. For certain patterns found in the OWL specification, verbalization patterns can be presented to the user. The verbalization process was also more sensible to word categories (e.g., nouns, verbs, adjectives). If an adjective is detected within an object property or within a restriction then the verbalization procedure considers this fact. This technique produces a better verbalization output.

Graphical visualization was included in this paper since it was the intention to show that a representation of an ontology must be a mix of different visualization techniques. It must be left to the user to choose the right representation for a certain situation.

### IX. CONCLUSION AND FUTURE WORK

In this paper, a visualization approach and its implementation was described. In order to browse a network of ontologies it was proposed that different visualization strategies must be combined together. It is also necessary to provide the user with a view of interrelated documents and to help him during his navigation through the ontology document network. The prototype was implemented in Java. The Jena API was used for OWL ontology parsing. For the graphical visualization JGraph (www.jgraph.com) was used.

The intention also leaves many future research and implementation issues. First of all, it could be interesting to integrate also three-dimensional visualization views. Since there are also a lot of documents in other formats (e.g.,

RDF) it would be good to extend the functionality in such a way that also RDF documents are visualized. Finally an integration of this approach to Protégé is worth to look at. For users, which are more familiar with mathematical expressions and logic, a detailed view of the logical statements expressed in OWL could be interesting

## REFERENCES

- [1] Ch. Kop, "How to Summarize an OWL Domain Ontology," ICDS 2010, Proc. of the International Conf. of the Digital Society, IEEE Computer Society Press, 2010, pp. 106 – 111.
- [2] D. Bezerra, A. Costa, and K. Okada, "SwTOI (Software Test Ontology Integrated) and its applicaton in Linux Test," Proc. of the 3rd Int. Workshop on Ontology, Epistemology and Conceptualization for Information Systems, Software Engineering and Service Science, CEUR-WS, Vol 460, 2009, <http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/>, 19.01.2011.
- [3] K. Bontcheva, "Generating Textual Summaries from Ontologies," Proc. of the Second European Semantic Web Conference, ESWC 2005, Lecture Notes in Computer Science (LNCS), Vol. 3532, Springer Verlag, Berlin, Heidelberg, 2005, pp. 531 – 545.
- [4] G. Fliedl, C. Kop, and J. Voehringer "From OWL class and property labels to human understandable natural language," Proc. of the 12th International Conference on Applications of Natural Language to Information Systems, NLDB 2007, Lecture Notes in Computer Science (LNCS), Vol. 4592, Springer Verlag, 2007, pp. 156 – 167.
- [5] N. E. Fuchs, S. Höfler, K. Kaljurand, F. Rinaldi, and G. Schneider, „Attempto Controlled English: A Knowledge Representation Language Readable by Humans and Machines,” Reasoning Web, First International Summer School, LNCS 3564, Springer, 2005, pp. 213-250.
- [6] D. Hewlett, A. Kalyanpur, V. Kolovski, and C. Halaschek-Wiener, „Effective Natural Language Paraphrasing of Ontologies on the Semantic Web,” End User Semantic Web Interaction Workshop, CEUR-WS Proceedings, Vol. 172, 2005, <http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/>, 19.01.2011.
- [7] N. Huang and Sh. Diao, "Structure-Based Ontology Evaluation," IEEE International Conference on e-Business Engineering (ICEBE06), 2006, pp. 1- 6.
- [8] K. Kaljurand and N.E. Fuchs, "Verbalizing OWL in Attempto controlled English," Proc. of the 3rd Int. Workshop on OWL Experiences and Directions (OWLED2007), CEUR-WS Proceedings, Vol. 258, 2007, <http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/>, 19.01.2011.
- [9] Ch. Kop, "What are main concepts in an OWL domain ontology," 1st Int. Conference on Knowledge Engineering and Ontology Development, INSTICC Proceedings, Funchal, 2009, pp. 404 – 407.
- [10] D.L. McGuinness and F. van Harmelen., "OWL Web Ontology Language Overview," <http://www.w3.org/TR/owl-features/>, 2004, 19.01.2011.
- [11] C. Mellish and X. Sun, "The Semantic Web as a Linguistic Resource: Opportunities for Natural Language Generation," Knowledge Based Systems Vol. 19, 2006, pp. 298-303.
- [12] D.L. Moody, "Entity Connectivity vs. Hierarchical Levelling as a Basis for Data Model Clustering: An Experimental Analysis," DEXA 2003 Proceedings, Lecture Notes in Computer Science (LNCS), Vol. 2736, Springer Verlag, Berlin, Heidelberg, 2003, pp. 77-87.
- [13] D.L. Moody and A. Flitman, "A Methodology for Clustering Entity Relationship Models – A Human Information Processing Approach," Proc. of Conceptual Modeling (ER 1999), Lecture Notes in Computer Science (LNCS), Vol. 1728, Springer Verlag, Berlin, Heidelberg, 1999, pp. 114-130.
- [14] N. Shadbolt, W. Hall, and T. Berners-Lee, "The Semantic Web Revisited," IEEE Intelligent Systems Journal, May/June 2006, pp. 96-101.
- [15] <http://krono.act.uji.es/Links/ontologies/>, 19.01.2011
- [16] M. Lanzemberger, J. Sampson, and M. Rester, "Visualization in Ontology Tools," International Conference on Complex, Intelligent and Software Intensive Systems. IEEE Computer Society Press, 2009, pp. 705 – 711.
- [17] OntoViz: Visualizing protégé ontologies, <http://protegewiki.stanford.edu/index.php/OntoViz>, 15.08.2011
- [18] M. Horridge, OWLViz <http://www.co-ode.org/downloads/owlviz/OWLvizGuide.pdf>, 19.01.2011
- [19] P. Eklund, N. Roberts, and S.Green "Ontorama: Browsing RDF Ontologies using a hyperbolic-style browser," Proc. of 1<sup>st</sup> Intl. Symposium on CyberWorlds, 2002, pp. 405 – 411.
- [20] A. Bosca and D. Bonino, "Ontosphere: more than a 3d ontology visualization tool," SWAP 2005 the 2<sup>nd</sup> Italian Semantic Web Workshop, 2005, <http://ceur-ws.org/Vol-166/70.pdf>, 19.01.2011
- [21] S. Decker, M. Erdmann, D. Fensel, and R. Studer "Ontobroker: Ontology based access to distributed and semi structured information," Database Semantics: Semantic Issues in Multimedia Systems, Kluwer Academic Publisher, 1999, pp. 351 – 369.
- [22] M. Lanzemberger, S. Miksch, and M. Pohl, "The stardines – visualizing highly structured data," Proc. of the International Conf. on Information Visualization, IEEE Computer Society Press, 2003, pp. 47 – 52.
- [23] M. Lanzemberger and J. Sampson, "AIViz – A Tool for Visual Ontology Alignment," Proceedings of the Information Visualization (IV'06), IEEE Computer Society Press, 2006, pp. 430 – 440.
- [24] S.M. Falconer, R. I. Bull, L. Grammel, and M-A. Storey, "Creating visualizations through ontology mapping," International Conference on Complex, Intelligent and Software Intensive Systems, IEEE, 2009, pp.688-693
- [25] O. Gilson, N. Silva, P.W. Grant, and M. Chen, "From Web Data to Visualization via Ontology Mapping," Eurographics /IEEE-VGTC Symposium on Visualization , Vol. 27, No. 3, 2008. pp. 959-966.
- [26] J. Garcia, F. Garcia, R. Theron, "Visualising semantic coupling among entities in an OWL ontology," appears in Proc of the 4th International Workshop on Ontology, Empistemology and Conceptualization for Information Systems, Software Engineering and Service Science, Springer Verlag, Vol. 62.
- [27] C. Bezerra, F. Freitas, J. Euzenat, A. Zimmermann, "ModOnto: A tool for modularizing ontologies," CEUR-WS, Vol. 427, <http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-427/paper3.pdf>, 19.01.2011
- [28] V. Sugumaran, S. Puroa, V.C. Storey, and J. Conesa, "On-Demand Extraction of Domain Concepts and Relationships from Social Tagging Websites," Proc. of the 15th Int. Conference on Natural Language Processing and Information Systems, Lecture Notes in Computer Science (LNCS), Vol. 6177, Springer Verlag, 2010 pp. 224 – 232.
- [29] S. Peroni, E. Motta, and M. d'Aquin, "Identifying key concepts in an ontology, through the integration of cognitive principles with statistical and topological measures," Proc. of the 3rd Asian Semantic Web Conference on the Semantic Web (ASWC'08), Lecture Notes in Computer Science (LNCS), Vol. 5367, Springer Verlag, 2008, pp. 227 – 241.
- [30] J. Vrana and M. Mach, "Ontology concepts interpretation," 8th International Symposium on Applied Machine Intelligence and Informatics, IEEE Press, 2010, pp. 215 – 219.
- [31] X. Zhang, G. Cheng, and Y. Qu, "Ontology Summarization Based on RDF Sentence Graph," Proceedings of the 16th International Conference on World Wide Web (WWW 2007), ACM, NY, , 2007, pp. 707 – 715, doi 10.1145/1242572.1242668.
- [32] [www.ontologydesignpatterns.org](http://www.ontologydesignpatterns.org), 19.01.2011