

OntoCAT: An Ontology Consumer Analysis Tool and Its Use on Product Services Categorization Standards

Valerie Cross and Anindita Pal

Computer Science and Systems Analysis, Miami University,
Oxford, OH 45056
{crossv, pala}@muohio.edu

Abstract. The ontology consumer analysis tool, OntoCAT, provides a comprehensive set of metrics for use by the ontology consumer or knowledge engineer to assist in ontology evaluation for re-use. This evaluation process is focused on the size, structural, hub and root properties of both the intensional and extensional ontology. It has been used on numerous ontologies from varying domains. Results of applying OntoCAT to two Product and Service Categorization Standards, UNPSCS and ecl@ss ontologies are reported.

Keywords: ontology evaluation, ontology metrics, ontology ranking, hubs

1 Introduction

The need for domain ontology development and management is increasingly more and more important to most kinds of knowledge-driven applications. Development and deployment of extensive ontology-based software solutions represent considerable challenges in terms of the amount of time and effort required to construct the ontology. These challenges can be addressed by the reuse of ontologies. The extent to which reuse of ontologies could contribute cost and time savings parallels that obtained in software reuse [17] because acquiring domain knowledge, constructing a conceptual model and implementing the model require a huge effort. As with any other resource used in software applications, ontologies need to be evaluated before use to prevent applications from using inconsistent, incorrect, or redundant ontologies. Even if an ontology has none of these problems and is formally correct, users must decide whether the content of ontology is appropriate for the requirements of their project, that is, they must determine how well the metadata and instances meet the requirements for their problem domain. Knowledge engineers need an ontology analysis tool to help in the process of ontology assessment for reuse.

Much ontology research has focused on new methodologies, languages, and tools [4]; recently, however, since the OntoWeb 2 position statement stressed the insufficient research on ontology evaluation and the lack of evaluation tools [11], much attention has been directed towards ontology evaluation [3,8]. Initially this attention concentrated on a formal analysis approach to evaluating ontologies [9]. Others have created taxonomies of ontology characteristics [12] to quantify the

suitability of ontologies for users' systems. Knowledge engineers must analyze these characteristics for the prospective ontologies in order to compare them and select the appropriate ontology for the system. More recent efforts address evaluating ontologies for reuse, not by ontology developers and experts, but by ontology consumers [14] who are users such as system project managers hoping to find existing ontologies on the Semantic Web which can be reused and adapted for their systems.

The objective of this research is to describe an ontology consumer analysis tool, OntoCAT [16], that summarizes essential size, structural, root and hub properties for both an intensional ontology and its corresponding extensional ontology. An intensional ontology only includes the ontology schema or definitions. An extensional ontology includes the instances, i.e., occurrences of classes and relationships. OntoCAT supports the ontology consumer by performing an analysis on the graph-like properties of an ontology. First, a brief overview of the variety of approaches to evaluating ontologies is presented in Section 2. Included in more detail in this presentation are those methods which take the ontology consumer perspective on evaluation. Section 3 describes some of the metrics included in OntoCAT. OntoCAT has been created as a plug-in for the Protégé Ontology Editor (<http://protege.stanford.edu/overview/>). The OntoCAT user interface is presented in Section 4. The results of performing an ontology consumer analysis on two product services and categorization standards (PSCS), UNSPSC (United Nations Standard Products and Services Code) and eCl@ss are discussed in Section 5 along with a brief description of the UNSPC and eCl@ss ontologies. Conclusions and future planned work are presented in Section 6.

2 Ontology Evaluation

A variety of approaches to ontology evaluation have been proposed depending on the perspectives of what should be evaluated, how it should be evaluated and when it should be evaluated. As such, ontology evaluation has become a broad research area [3] with numerous frameworks proposed for evaluating how "good an ontology is". These frameworks can be classified into various categories depending on what qualities are considered most relevant to an ontology and how they are being evaluated. These qualities may also have an importance factor. For example, is the quality of the design more important than the quality of the content [15] and can a gold standard be used or is an assessment by a human expert required [3]? In addition some evaluation methods make specific recommendations about when in the ontology development lifecycle the evaluations should be performed. Others suggest developing methodologies to evaluate an ontology during the development process and throughout its entire lifetime [9].

Another distinction made in ontology evaluation is that of selection versus evaluation. In [18], ontology selection is defined as "ontology evaluation in the real Semantic Web." Their survey of existing selection algorithms reveals that few ontology evaluation methods are incorporated except for similarities in topic coverage. They conclude that although evaluation and selection consider different requirements, they are complementary. In [7] a holistic view of ontology evaluation

is considered by viewing an ontology as a communication object. The Qood grid method permits parametric design for both evaluation and selection (diagnostic) tasks. Ontologies are analyzed in their graph and formal elements, functional requirements, and annotation profile. The Qood evaluation based on graph analysis parallels that of the OntoCAT metrics proposed in [5].

Due to space limitations, not all of these various evaluation methods can be discussed. The following sections briefly describe ontology consumer evaluation methods and overview the two tools most closely related to OntoCAT.

2.1 Ontology Evaluation from the Consumers' Perspective

To make reusing ontologies easier, more research needs to address the evaluation of an ontology from the consumer point of view [14]. Ontology consumers need tools to help with two different tasks, selecting from the enormous number of available ontologies the most appropriate ontology candidates for their applications and quality evaluation of ontologies. As pointed out previously, ontology evaluation and ontology selection are complementary. The question is in what order should these two tasks be performed. The answer might depend on the individual ontology developer, but typically the answer is that selection is performed first for filtering purposes and then followed by a more time consuming quality evaluation. Selection or filtering methods typically employ topic coverage, popularity, and richness of conceptualized knowledge [18].

Consumer analysis tools could be useful to both selection and evaluation tasks. One approach suggested in [14] for consumer analysis is ontology summarizations. Ontology summarizations are analogous to approaches used by researchers in reviewing the usefulness of papers or deciding on whether to purchase a book. Just as a researcher examines a book's summary or a paper's abstract when deciding on its usefulness, similarly there should be some abstract or summary of what an ontology covers to help consumers decide if it fits their application requirements. The summary might include the top-level concepts and links between them as a graphical representation and a listing of hub concepts – concepts that have the largest number of links in and out. It could also include metrics similar to Google's page rank that determine that a concept is more important if other important concepts are linked to it.

OntoCAT metrics are based on these analogies and fall into both the structural and functional types of measures for ontology evaluation [7]. OntoCAT metrics can be valuable to both the selection and evaluation tasks performed on ontologies. The summaries which provide the consumer with a high level view of the topic coverage are functional types of measure and important to the selection task. The OntoCAT metrics analyzing an ontology as a graph structure are structural metrics that can be used to evaluate the quality of the ontology design similar to those used for software metrics [17]. Two of the more recent related approaches to OntoCAT are presented below. The ones selected either focus on quantitative size and structural metrics for ontology selection or evaluation or they have a component that includes such metrics. Structural types of measures in [7] correspond closely with OntoCAT metrics presented in [5].

2.2. OntoQA

The OntoQA tool [20] developed by LSDIS Lab at the University of Georgia measures the quality of ontology from the consumer perspective. The OntoQA tool measures the quality of the ontology in terms of Schema and Instance metrics. The schema metrics of OntoQA address the design of the ontology schema, some of which correspond with the OntoCAT intensional metrics calculated on the intensional ontology (the definition of the ontology). Instance metrics of OntoQA deal with the size and distribution of the instance data, some of which correspond with the OntoCAT extensional metrics calculated on the ontology occurrences.

OntoQA defines three metrics in its Schema metrics group. These are relationship richness, attribute richness and inheritance richness. Relationship richness measures the percentage of relationships that are not is-a or taxonomic relationships. Attribute richness measures the average number of attributes per class, dividing the cardinality of the attribute set by the cardinality of the class set. Inheritance richness measures the average number of subclasses per class.

Metrics for the Instance group are categorized into two subclasses: the whole instance ontology or class metrics. Class metrics are used to describe how a class is being populated in the instance ontology. Metrics for the whole instance ontology include class richness, average population, and cohesion. Class richness measures the distribution of instances across classes. Formally, it is defined as the ratio of the number of classes with instances divided by the number of classes defined in the ontology. Average population measures the average distribution of instances across all classes. Formally, it is defined as the number of instance divided by the number of class in the ontology. Cohesion measures the number of connected components in the instance graph built using the taxonomic relationships among instances.

Class metrics include importance, fullness, inheritance richness, connectivity, relationship richness and readability. Importance refers to the distribution of instance over classes and is measured on a per sub-tree root class. It specifies the percentage of instances that belong to classes in the sub-tree rooted at the selected class with respect to the total number of instances in the ontology. This definition is somewhat confusing because multiple instance sub-trees for a selected class could exist. It is assumed that this definition would include all instances of sub-trees with the selected class type. Fullness is primarily for use by ontology developers to measure how well the data population was done with respect to the expected number of instances of each class. It is similar to importance except that it is measured relative to the *expected* number of instances that belong to the sub-tree rooted at the selected class instead of the total number of instances in the ontology. Connectivity for a selected class is the number of instances of other classes connected by relationships to instances of the selected class. Relationship richness measures the number of the properties in the selected class that are actually being used in the instance ontology relative to the number of relationships defined for the selected class in the ontology definition. Readability measures the existence of human readable descriptions in the ontology. Human readable descriptions include comments, labels, or captions.

2.3 AKTiveRank

Several ontology search engines such as Swoogle [6] and OntoSearch [21] can be used by entering specific search terms to produce list of ontologies including search terms somewhere in the ontology. AKTiveRank [1] ranks the ontologies retrieved by an ontology search engine. Its initial implementation evaluated each retrieved ontology using four measures: class match, density, semantic similarity and centrality.

The class match measure evaluates the coverage of an ontology by providing a score based on the number of query terms contained in the ontology and evaluates using both exact match where the search term is identical to the class name and partial match where the search term is contained within the class name. The density measure evaluates the degree of details in the representation of the knowledge concerning the matching classes. The density value for an individual class is the weighted sum of the count of its number of super-classes, subclasses, direct and indirect relations (in and out), and siblings. The number of instances was initially included but dropped since it was felt that this parameter might bias evaluation toward populated ontologies [2]. This bias might penalize ontologies with higher quality definitions (schemas). The density measure for the query is the average for all matching classes. The semantic similarity measure (SSM) determines how close the classes that match the search terms are in an ontology. The semantic similarity is calculated between all pairs of matching classes and then the average is taken.

The centrality measure assumes that the more central a class is in hierarchy, the better analyzed and more carefully represented it is. A centrality measure for a class is calculated for each class that matches fully or partially a given query term based on its distance from the class midway from the root to the leaf on the path containing the matching class. Then the centrality measure for the query is the average for all matching classes. More recent research [2] identified the redundancy of the centrality measure because of its close correspondence with the density measure and replaced it with the betweenness measure. The higher the betweenness measure for a class, the more central that class is to the ontology. For betweenness, the number of shortest paths between any two classes that contains a class matching a queried concept is calculated. These numbers are summed over all queried concept. Their average determines the betweenness measure. The overall rank score for an ontology is the weighted aggregation of these resulting component measures is performed to produce the overall rank of the ontology.

The researchers creating AKTiveRank note that such a tool “needs to be designed in a way so as to provide more information of the right sort. Mere ranking, while it will no doubt be useful, may not be the whole story from a practical perspective” and further suggest that there is “a need to disentangle a number of different parameters which can affect the usability of a knowledge representation” since the perception of the knowledge engineers with respect to different ontology parameters “may vary depending on the particular application context” [1]. A limitation of this tool is that it only ranks intensional ontologies since all measurements are based on the definition of the ontology. There are some ontologies, especially terminological ontologies, whose intensional ontology is quite simple but whose extensional ontology is quite complex. An ontology consumer analysis tool should be able to process both components of an ontology to provide the user with as much information as possible.

3 OntoCAT Metrics

The ontology consumer analysis tool OntoCAT provides a comprehensive set of metrics to be used by the ontology consumer or knowledge engineer to assist in ontology evaluation for re-use. Quality measurements are not being provided but instead summarizations, size and structural metrics are provided. The metrics are separated into two categories: intensional metrics and extensional metrics. Intensional metrics are calculated based on the ontology definition itself, that is, its classes and subclasses and their properties. Extensional metrics measure the assignment of actual occurrences of ontological concepts, that is, the instances and how effectively the ontology is used to include the domain knowledge. Much research has been focused on extensional ontologies, in some part, because the consideration for reuse of ontologies has often been on terminological ontologies such as found in the biomedical fields.

The following metrics are relative to the metadata being assessed, C – class, P – property, A – attribute, and R – relationship. Metrics beginning with an “i” are for the intensional ontology, and those beginning with an “e” are for the extensional ontology. Some of the metrics do not return a numeric value but instead indicate identifying information. For example, iMaxClasses provides a list of classes that have the maximum number of properties. In the following, Cnt stands for count, Avg for average, and Rng for range. The main approach is to determine various metrics and to also examine them on both horizontal (per depth) and vertical (per path) slices of the ontology. Below only a sample of the metrics are presented due to page limitations. A complete description of all the metrics can be found in [16].

3.1 Size Metrics

Intensional. Typically an intensional ontology has one root concept, but multiple root concepts are possible. If no concept or class c_j is specified, the intensional size metric is calculated for the entire ontology, that is, over all the trees defined in the ontology. When a concept c_j is specified to be used as a root, the size metric is calculated on the tree specified by the selected concept c_j as its root. Although the measures using a root are referred to as size metrics, they do, however, use the “*is-a*” or subsumption hierarchy to determine the tree for which the size metrics are being determined.

iCnt(C) = the number of classes for the entire intensional ontology

iCnt(C)(c_j -root) = the number of classes for the sub-tree at the selected class c_j .

iCnt(P) = the number of properties defined for the entire intensional ontology

iCnt(P)(c_j -root) = the number of properties defined for the entire sub-tree at class c_j . A property may be inherited from its parents. Only new properties are counted for each class.

iCntTotal(P)(c_j) = the total (new + inherited) number of properties for class c_j .

iCnt(R) = the number of relationships defined for the entire intensional ontology. A relationship is a special kind of property that has a class as its range.

iCntTotal(R)(c_j) = the total (new + inherited) number of relationships defined for only the selected class

$iMaxTotal(P \text{ to } C)(c_j\text{-root}) = \max$ number of (new + inherited) properties defined for a single class over all classes in the sub-tree at the selected class c_j
 $iMaxTotalClasses(P \text{ to } C)(c_j\text{-root}) =$ class names for classes within sub-tree at the selected class c_j that have the max number of properties

Extensional.

$eCnt(c_j) =$ the number of object occurrences for class c_j
 $eCnt(C) = \sum_j eCnt(c_j)$, the total number of object occurrences in the ontology
 $eCnt(C)(c_j\text{-root}) = \sum_i eCnt(c_i)$, the total number of object occurrences in the sub-trees for the selected class c_j where c_i is in sub-tree c_j
 $eAvgCnt(C) = eCnt(C)/iCnt(C)$, the average number of occurrences for all classes
 $eMaxCnt(C) = \max_i[eCnt(c_i)]$ and identify $eMaxCntClass$, i.e., the class with the maximum number of occurrences in the ontology
 $eCnt(r_i) =$ the number of occurrences for relation r_i
 $eCnt(R) = \sum_i eCnt(r_i)$ total number of occurrences for all relations in ontology
 $eAvgCnt(R) = eCnt(R)/eCnt(C)$, average number of relationships per occurrence
 $eMaxCnt(R) = \max_i[eCnt(r_i)]$ and identify $eMaxCntRelation$

3.2 Structural Metrics

Intensional structural metrics are similar to size metrics since they are over the entire intensional ontology, that is, over all the root trees defined in the ontology if no concept or class is specified. When a class is specified, the structural metrics are calculated for the entire sub-tree at that class c_j . The class hierarchy (sub-class/super-class) relationships are used for the structural organization.

$iCnt(Roots) =$ number of roots in the ontology.
 $iCnt(leaves(c_j\text{-root})) =$ number of leaf classes of the sub-tree at the selected class c_j
 $iCnt(leaves) = \sum_j iCnt(leaves(c_j\text{-root}))$; the total number of leaf classes in the entire ontology where $c_j\text{-root}$ is a root class.
 $iPer(leaves(c_j\text{-root})) = iCnt(leaves(c_j\text{-root})) / iCnt(C)(c_j\text{-root})$ the fraction of classes that are leaves of the is-a hierarchy for the entire sub-tree at class c_j .
 $iAvg(leaves) = iCnt(leaves)/iCnt(C)$, the fraction of classes that are leaves for the entire ontology.

$iMaxDepth(c_j\text{-root}) = \max_j [\text{depth}(\text{leaf}_{ij})]$, the maximum depth of the sub-tree at the selected class c_j and return the class of the leaf at the maximum depth

Several intensional structural metrics are adapted from WordNet's (IC) measure [19]. The IC for class c_{ij} for $c_j\text{-root}$ (the class may be in multiple trees, therefore, the subscript j specifies the root of the tree) is given as [5]:

$$iIC(c_{ij}) = 1 - \log(iCnt(C)(c_{ij}\text{-root}) + 1) / \log(iCnt(C)(c_j\text{-root}))$$

The class $c_j\text{-root}$ must be a root class of the ontology whereas, $c_{ij}\text{-root}$ is any class within the inheritance hierarchy rooted at $c_j\text{-root}$. This measure can be used to identify the degree of information content on a per depth basis within the ontology. Using class information content as a new measure provides a novel way of examining the ontology for symmetry and balance across its horizontal slices. Some of the following measures proposed for each $c_j\text{-root}$ of an intensional ontology are:

$$\begin{aligned}
iIC(\text{depth}_k(c_j\text{-root})) &= \sum_i IC(c_{ij}) \text{ for all } c_i \text{ at depth } k \text{ for } c_j\text{-root} \\
iAvgIC(\text{depth}_k(c_j\text{-root})) &= iIC(\text{depth}_k(c_j\text{-root})) / iWidth(\text{depth}_k(c_j\text{-root})) \\
iRngIC(\text{depth}_k(c_j\text{-root})) &= iMaxIC(\text{depth}_k(c_j\text{-root})) - iMinIC(\text{depth}_k(c_j\text{-root})) \\
iAvgIC(c_j\text{-root}) &= \sum_k iAvIC(\text{depth}_k(c_j\text{-root})) / iMaxDepth(c_j\text{-root}), \text{ the average IC} \\
&\text{over all depths in the tree at root } c_j\text{-root}
\end{aligned}$$

Extensional. Structural metrics are calculated on the specified root concept $c_j\text{-root}$ and the specified relationships used to create the structural organization of the extensional ontology. For example, in the WordNet extensional ontology, the specified relationships providing its structure are the hyponym and hypernym relationships. If no concept is specified or if the specified concept is the top most concept of the ontology, structural extensional metrics for the complete ontology are calculated. When $c_j\text{-root}$ is specified for metrics of extensional ontologies, only root occurrences of this class with respect to the structural organization of the extensional ontology are considered. The metrics listed below that have the occ parameter are automatically produced for each root occurrence of the class selected by the user.

$$\begin{aligned}
eCnt(\text{roots}) &= \text{number of root occurrences for all root classes} \\
eCnt(\text{leaves}(c_j\text{-root})) &= \text{number of leaves for all occurrences of class } c_j\text{-root}_i. \\
eCnt(\text{leaves}(c_j\text{-root}(occ))) &= \text{number of leaves for specified occurrence of } c_j\text{-root}_i. \\
eMaxCnt(\text{leaves}(c_j\text{-root})) &= \max_i [eCnt(\text{leaves}(c_j\text{-root}(occ_i)))], \text{ the maximum} \\
&\text{number of leaves in all rooted occurrences of class } c_j\text{-root, give its identity} \\
eMinDepth(c_j\text{-root}(occ)) &= \min_i [\text{depth}(\text{leaf}_{ij}(occ))], \text{ the minimum depth of the} \\
&\text{sub-tree at the selected root occurrence of root class } c_j \text{ and return the leaf} \\
&\text{occurrence(s) at the minimum depth.} \\
eWidth(\text{depth}_k(c_j\text{-root})) &= (\sum_i eWidth(\text{depth}_k(c_j\text{-root}(occ_i))), \text{ the number of} \\
&\text{instances at depth } k \text{ for all occurrences of the selected root class } c_j
\end{aligned}$$

3.3 Summarization Metrics

The hub summary displays information on the hubs, i.e., object occurrences (for extensional) and classes (for intensional) having the maximum number of links in and out. For intensional, the count of links is the number of subclasses and superclasses defined. For extensional, the links are based on the relationships specified for creating its structure. A list of the top n hubs (user-specified), is reported with statistics for each hub. Note that the 'i' or 'e' preceding the metric is omitted since it is determined by whether it is for an intensional or extensional ontology.

$$\begin{aligned}
\text{depth}(\text{hub}) &= \text{the depth of the tree where the hub concept is located} \\
\text{width}(\text{hub}) &= \text{the number of other occurrences at that depth in the tree} \\
\text{Cnt}(P(\text{hub})) &= \text{number and list of properties defined for the hub in case of classes} \\
\text{Cnt}(\text{child}(\text{hub})) &= \text{the number and list of direct children of the hub} \\
\text{Cnt}(\text{parent}(\text{hub})) &= \text{the number and list of direct parents of the hub}
\end{aligned}$$

A root summary may be calculated for both the intensional and extensional ontology and include class or occurrence counts for roots and leaves, the minimum, maximum and average depths of the intensional and extensional ontology, and the minimum, maximum, and average widths of the intensional and extensional ontology.

4 OntoCAT User Interface

OntoCAT is implemented as a Protégé plug-in so that it is incorporated into the ontology engineering environment itself. As an ontology is being developed, OntoCAT may be executed to determine how the structural types of measures change during the development cycle. Since OntoCat is part of the ontology engineering environment, evaluation can easily be performed without altering the ontology.

The implementation is generalized to handle the structural difference in ontologies and is parameterized to permit easily switching between an intensional and extensional ontology. The user selects the root class and relationship to be used to calculate the metrics. The implementation uses the OWL API because of its flexibility and easy access for ontologies. Metrics for ontologies in RDF/RDF(S) can also be calculated through conversion to OWL ontologies with Protégé's export function.

The main user interface consists of two split panes. The left "Selection" panel permits selection of the metrics. The "Result" panel displays the calculated metrics. In the following figures, a small version of the WordNet ontology is used. WordNet is a general terminological ontology of the English language which serves as a freely available electronic dictionary organized by nouns, verbs, adjectives and adverbs into synonym sets (synsets), each representing one underlying lexical concept [13].

4.1 Selection Panel

The user selects which metrics to calculate for the ontology. Figure 1 shows the "Selection" panel. The metrics are grouped into intensional size and structure and extensional size and structure. This organization allows the users to easily switch between the intensional and extensional ontology. The IC metrics are separated from the structural metrics for aesthetic reasons. Users also enter the depth values in the two text fields for calculating the IC metrics and the width metrics at the depth.

The next set of parameters to input after selection of metrics is the root concept on which to measure and the relationship used to build the extensional taxonomical structure. The user can select these parameters by clicking the "Metrics" or "Report" buttons. When these buttons are clicked a pop-up window is opened as shown in Figure 2 below. This window contains the list of classes and relationships defined in the ontology. For example, after selecting the metrics shown in Figure 1, the user clicks on the "Metric" button. The concept and relation selection window pops up. In Figure 2, the Lexical Concept class of the WordNet ontology is selected. By default metrics are calculated on the whole ontology so that users only need to select the class on which they want to calculate metrics. For the "Metric" button if no class is selected, then only the ontology level metrics are displayed since there is no space in the UI to display the metric results for all classes in the ontology. For the "Report" button, if no class is selected, the metrics are calculated on all classes in the ontology. The "Report" button generates a report of the selected metrics to a file in the tools home directory. This report is formatted for easy importing to an Excel spreadsheet to analyze and generate charts and tables as done in Section 5. If users do not select a class, the report is generated on each of the classes in the ontology.

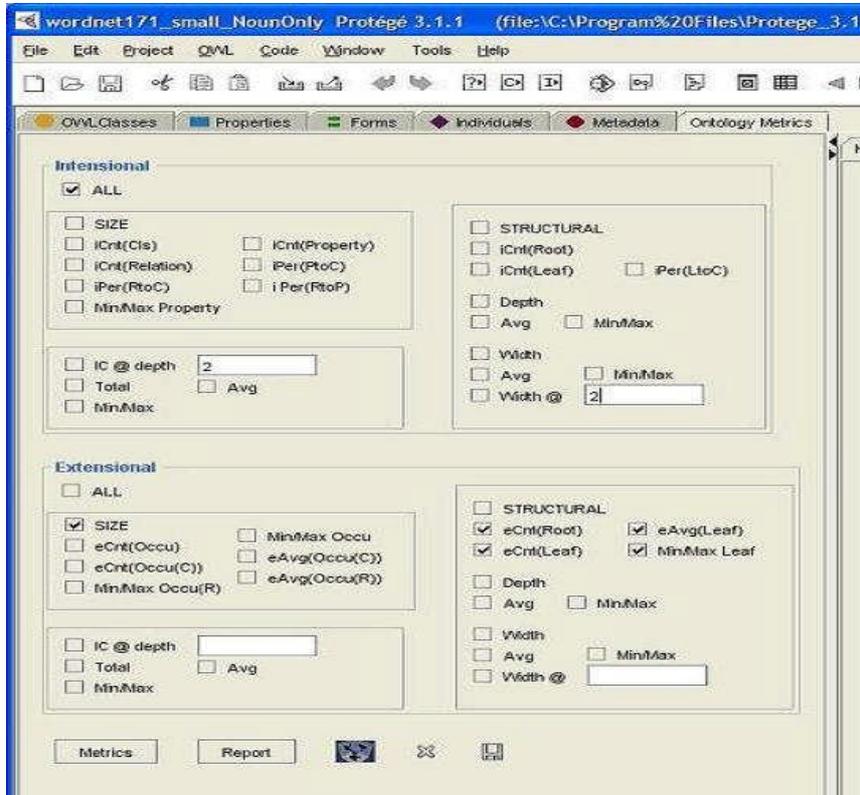


Fig. 1. Selection Panel with list of metrics

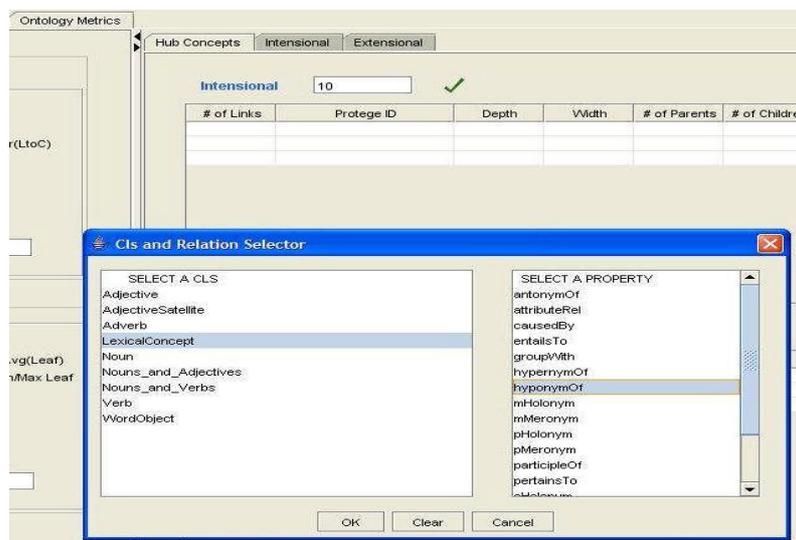


Fig. 2. Selection of Class and Relationship

The user selects the relationship for building the extensional taxonomic structure. For intensional, the structure uses the sub-class relationship. The extensional taxonomic structure differs from ontology to ontology. For example, WordNet uses the hyponymOf and hypernymOf relationships. If the extensional metrics button is selected, the parent relationship for structuring must be entered.

4.2 Result Panel

The “Result” panel displays the metrics from the “Selection” panel and has three tabs: Hubs, Intensional, and Extensional. Figure 3 shows the hub summary for a small WordNet ontology. The intensional table lists the hub classes, those with the maximum number of subclasses and super-classes. The extensional table lists the instance hubs, those with the maximum number of links in and out. The last table is specific to the class listed above the table. For example, the third table lists the extensional hubs for the lexicalConcept class selected in the “Selection” panel pop-up window. The summary provides the following: depth, width, number of properties and IC. By default the plug-in displays the top 10 hub concepts. Users can specify the number or percent of hubs to display by changing the value in the text fields (for example to 20 or 10%), located beside the table labels and clicking the ✓ button.

5 Analysis of UNSPSC and ecl@ass Ontologies

An important requirement of e-Commerce is effective communication between software agents. A common approach to provide machine-accessibility for agents is a standardized vocabulary of product and services terminology referred to as Product and Service Categorization Standards (PSCS) [10]. UNSPSC (United Nations Standard Products and Services Code) and eCl@ss are two example PSCS developed into intensional ontologies consisting of the schemas and definitions of the concepts in the product and service domain. UNSPSC is a hierarchical classification of all products and services for use throughout the global marketplace. eCl@ss, developed by German companies, offers a standard for information exchange between suppliers and their customers. It uses a 4-level hierarchical classification system that maps the market structure for industrial buyers and supports engineers at development, planning and maintenance. Martin Hepp has developed an OWL version (<http://www.heppnetz.de/eclassowl>).

A previous study of PSCS ontologies uses a framework of metrics “to assess the quality and maturity of products and services categorization standards” [10]. This framework is applied to the most current and multiple past releases of eCl@ss, UNSPSC, eOTD, and RNTD. In that study, the term “size of segments” corresponds to OntoCAT’s $iCnt(C)(c_j\text{-root})$, the number of classes for a root class. The term “size” corresponds to OntoCAT’s $iCnt(C)$, the number of classes for the entire intensional ontology. The “property list size” corresponds to $iCnt(P)$, the number of properties defined for the entire intensional ontology. Using OntoCAT, an analysis for both

UNSPSC and eCI@ss was performed. Due to space limitations, only root summary reports are provided below in table format. Because eCI@ss has over 25000 roots, its root summary shows only a selected set of roots that have more interesting data.

Table 1 shows the root summary for the UNPSCS ontology. It is arranged in descending order of the total classes under that root class. Only the top 13 roots are shown due to space limitation. For all root classes there is a uniform maximum and minimum depth of 3. The root classes have all leaves at the same level and the maximum width occurs at the maximum depth, i.e., it is equivalent to the number of leaves for the root class. The minimum width varies but it always occurs at depth 1, i.e., the first level down from the root. The four root classes with the greatest number of classes and leaves are “Drugs Pharmaceutical_Products”, “Chemicals including Bio Chemicals and Gas Materials”, “Laboratory and Measuring and Observing and Testing Equipment”, and “Structures and Building and Construction and Manufacturing Components and Supplies.”

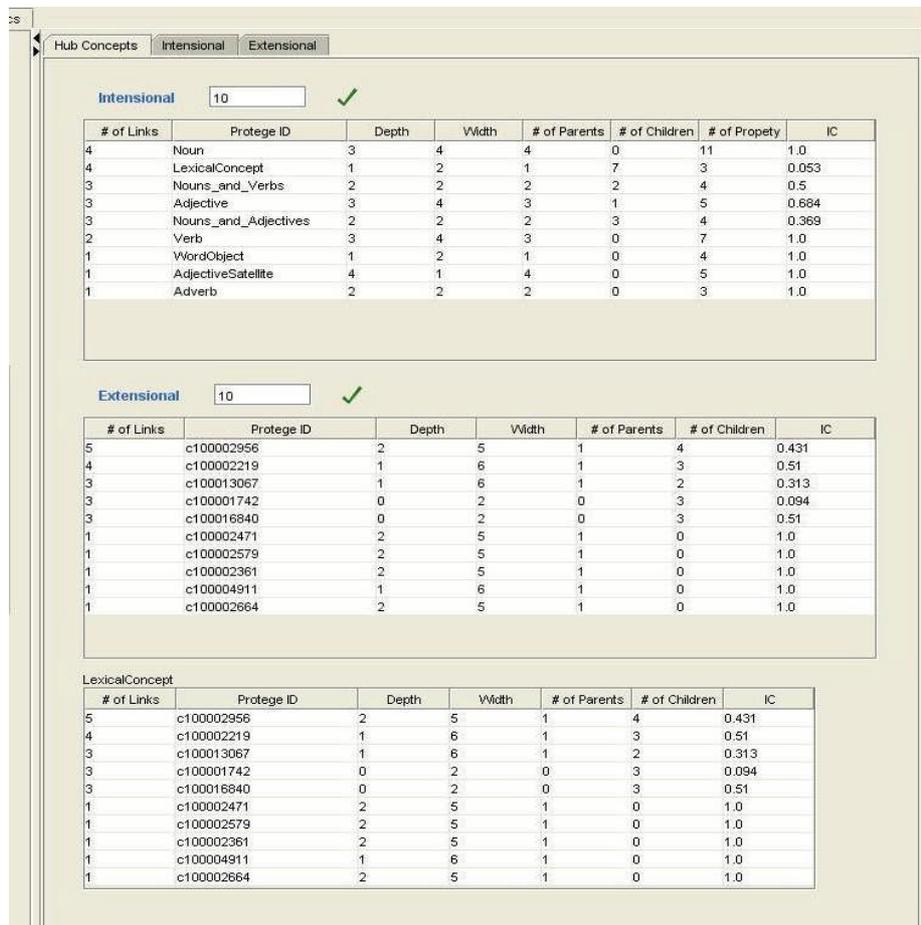


Fig. 3. The Result Panel showing Hub concept report

Table 2 displays the root summary for the several of the *_tax roots of the ecl@ss ontology. Note that the maximum depth for all the root classes is 4 and the minimum depth is one. Unlike UNPSCS, the length of every path for each root class in the ontology is not identical since a variation exists in the average depth. The maximum width occurs not at the greatest depth but at depth 3 for all roots. Like UNPSCS, the minimum width varies but is always occurs at depth 1 for each root. Looking at the ratio of total number of leaves to total number of classes, UNPSCS has a much larger percentage of leaf classes for its roots as compared to ecl@ss.

Table 1. UNPSCS Root Summary

Concept Name	Total Classes	Total Leaf	Max Depth	Min Depth	Avg Depth	Max Width	Level @ max	Min Width	level @ min	Avg Width
Laboratory_and_Measuring_and_Observing_and_Testing_Equipment	1103	1008	3.00	3	3	1008	3	3	1	367.66
Structures_and_Building_and_Construction_and_Manufacturing_Components_and_Supplies	696	615	3.00	3	3	615	12	1	1	232
Chemicals_including_Bio_Chemicals_and_Gas_Materials	614	508	3.00	3	3	508	3	14	1	204.66
Drugs_and_Pharmaceutical_Products	611	514	3.00	3	3	514	3	15	1	203.66
Commercial_and_Military_and_Private_Vehicles_and_their_Accessories_and_Components	496	417	3.00	3	3	417	3	10	1	165.33
Communications_and_Computer_Equipment_and_Peripherals_and_Components_and_Supplies	416	369	3.00	3	3	369	3	3	1	136.66
Industrial_Manufacturing_and_Processing_Machinery_and_Accessories	395	342	3.00	3	3	342	3	13	1	131.66
Distribution_and_Conditioning_Systems_and_Equipment_and_Components	340	312	3.00	3	3	312	3	4	1	113.33
Power_Generation_and_Distribution_Machinery_and_Accessories	305	275	3.00	3	3	275	3	5	1	101.66
Farming_and_Fishing_and_Forestry_and_Wildlife_Contracting_Services	280	237	3.00	3	3	237	3	8	1	93.33
Industrial_Production_and_Manufacturing_Services	279	236	3.00	3	3	236	3	9	1	93
Politics_and_Civic_Affairs_Services	279	241	3.00	3	3	241	3	8	1	93
Domestic_Appliances_and_Supplies_and_Consumer_Electronic_Products	275	246	3.00	3	3	246	3	7	1	91.66

Table 2. ecl@ss Root Summary

Concept Name	Total Classes	Total Leaf	Max Depth	Min Depth	Avg Depth	Max Width	Level @ max	Min Width	level @ min	Avg Width
C_AAG961003-tax	10623	5038	4	1	3.94	5292	3	20	1	2655.75
C_AAB572002-tax	5317	2181	4	1	3.8	2624	3	35	1	1329.25
C_AAB072002-tax	3983	1669	4	1	3.82	1973	3	19	1	995.75
C_AAD302002-tax	3585	1317	4	1	3.71	1756	3	37	1	896.25
C_AAF876003-tax	2927	1315	4	1	3.88	1444	3	20	1	731.75
C_AAC473002-tax	2653	1186	4	1	3.88	1320	3	7	1	663.25
C_AAC350002-tax	2431	1024	4	1	3.82	1192	3	24	1	607.75
C_AAB315002-tax	2127	850	4	1	3.77	1041	3	23	1	531.75
C_AAA183002-tax	2065	832	4	1	3.79	1019	3	14	1	516.25
C_AAA862002-tax	1927	739	4	1	3.73	932	3	32	1	481.75
C_AAA647002-tax	1603	589	4	1	3.68	763	3	39	1	400.75
C_AAD111002-tax	1519	580	4	1	3.74	750	3	10	1	379.75
C_AAF397003-tax	1451	499	4	1	3.62	680	3	46	1	362.75
C_AAT090003-tax	1239	445	4	1	3.64	577	3	43	1	309.75
C_AAD025002-tax	1041	318	4	1	3.57	502	3	19	1	260.25
C_AAW154003-tax	1007	417	4	1	3.79	490	3	14	1	251.75
C_AKJ313002-tax	977	403	4	1	3.79	477	3	12	1	244.25
C_AAD640002-tax	879	329	4	1	3.7	420	3	20	1	219.75
C_AKK397002-tax	863	286	4	1	3.62	416	3	16	1	215.75
C_AAC286002-tax	701	253	4	1	3.68	339	3	12	1	175.25
C_AAN560003-tax	515	214	4	1	3.8	253	3	5	1	128.75
C_AKJ644002-tax	509	121	4	1	3.41	242	3	13	1	127.25
C_AAE587002-tax	493	189	4	1	3.73	240	3	7	1	123.25
C_AAD170002-tax	451	175	4	1	3.74	221	3	5	1	112.75
C_AAC168002-tax	405	131	4	1	3.58	191	3	12	1	101.25

6 Summary and Conclusions

OntoCAT provides a comprehensive set of metrics for use by the ontology consumer. It may be used to assist in ontology evaluation for re-use or regularly during ontology development and throughout the ontology's lifecycle to record a history of the changes in both the intensional and extensional ontology. It includes either directly or components of many of OntoQA metrics. It differs from AKTiveRank which uses query concepts to rank ontologies. OntoCAT could be used to further analyze the top ranked ontologies produced by AKTiveRank. Numerous ontologies from varying domains: WordNet, UMLS, UNSPSC, and ecl@ss have been analyzed by OntoCAT. Here the results for the two PSCS ontologies have been reported. The metrics identified and implemented as plug-in software for Protégé are the most comprehensive set of metrics currently available in a tool for both kinds of ontologies. The tool still needs more capabilities to summarize the metrics both in intuitive terms and visually for the user. Another useful feature would be producing analysis based on query terms to provide a context on which to calculate more detailed metrics reflecting topic coverage. The structural types of metrics proposed in [7] that do not already exist in OntoCAT are to be further investigated for inclusion in OntoCAT.

References

1. Alani, H. and Brewster, C. Ontology Ranking based on the Analysis of Concept Structures, *International Conference On Knowledge Capture*, Alberta, Canada (2005)
2. Alani, H. and Brewster, C. Metrics for Ranking Ontologies, Fourth International Evaluation of Ontologies for the Web Workshop (EON 2006), Edinburgh, UK, May (2006).
3. Brank, J., Grobelnik, M., and Mladenic, D. A survey of ontology evaluation techniques. In *Proceedings of the 8th Int. multi-conference Information Society IS-2005*, 2005.
4. Corcho, O., Fernandez-Lopez, M., Gomez-Perez, A., Methodologies tools and languages for building ontologies, *Data & Knowledge Engineering* 46 41–64 (2003)
5. Cross, V. and Pal, A., Ontology Metrics, *2005 North American Fuzzy Information Processing Society*, Ann Arbor Michigan, July (2005)
6. Ding, L., T. Finin, A. Joshi, R. Pan, R. S. Cost, Y. Peng, P. Reddivari, V. C. Doshi, and J. Sachs. Swoogle: A semantic web search and metadata engine. In *Proc. 13th ACM Conf. on Information and Knowledge Management*, Nov. (2004)
7. Gangemi, A., Catenacci, C., Massimiliano, C. and Lehmann, J., Ontology Evaluation and Validation: An integrated formal model for the quality diagnostic task, On-line: http://www.loa-cnr.it/Files/OntoEval4OntoDev_Final.pdf (2005).
8. Gomez-Perez, A. Evaluating Ontology Evaluation, in Why evaluate ontology technologies? Because it works!, *Intelligent Systems, IEEE*, Volume 19, Issue 4, Jul-Aug 2004.
9. Guarino, N. and Welty, C. Evaluating ontological decisions with OntoClean, *Communications of the ACM*, Volume 45, Number 2, February(2002)
10. Hepp, M., Leukel, J., and Schmitz, V. A Quantitative Analysis of eClass, UNSPSC, eOTD, and RNTD: Content, Coverage, and Maintenance, *IEEE International Conference on eBusiness Engineering (ICEBE'05)* pp. 572-581 2005.
11. Kalfoglou, Y., Evaluating ontologies during deployment in applications, position statement, .OntoWeb 2 meeting 07-12-2002,
12. Lozano-Tello, A.and Gómez-Pérez, A., ONTOMETRIC: A Method to Choose the Appropriate Ontology, *Journal of Database Management*, 15(2), 1-18, April-June 2004.
13. Miller, G., WordNet: a lexical database for English, *Comm. of ACM* 38 (11), 39–41 (1995)
14. Noy, N., Evaluation by Ontology Consumers, in Why Evaluate ontology technologies? Because it works!, *IEEE Intelligent Systems*, July/August (2004)
15. Orbst, L., Hughes, T., and Ray, S. Prospects and Possibilities for Ontology Evaluation: The View from NCOR, Fourth International Evaluation of Ontologies for the Web Workshop (EON 2006), Edinburgh, UK, May (2006).
16. Pal, A. An Ontology Analysis Tool For Consumers, Masters Thesis, Miami University, Oxford, OH May (2006).
17. Poulin J.S. Measuring Software Reuse: Principles, Practices, and Economic Models. Addison Wesley Longman, (1997)
18. Sabou, M., Lopez, V, Motta, E. and Uren, V. Ontology Selection: Evlauation on the Real Semantic Web, Fourth International Evaluation of Ontologies for the Web Workshop (EON 2006), Edinburgh, UK, May (2006).
19. Seco, N.,Veale, T. and Hayes, J. An Intrinsic Information Content Metric for Semantic Similarity in WordNet, *ECAI 2004*, 1089-1090 (2004)
20. Tartir, S. Arpinar, I.B., Moore, M., Sheth, A.P. and Aleman-Meza, B. OntoQA: Metric-Based Ontology Quality Analysis, *IEEE Workshop on Knowledge Acquisition from Distributed, Autonomous, Semantically Heterogeneous Data and Knowledge Sources*, Houston, TX, USA, November 2005.
21. Zhang, Y., W. Vasconcelos, and D. Sleeman. Ontosearch: An ontology search engine. In *Proc. 24th SGAI Int. Conf. on Innovative Techniques and Applications of Artificial Intelligence*, Cambridge, UK (2004).