Ranking Ontologies in the Ontology Building Competition BOC 2014

Ioana Jimborean* and Adrian Groza*
* Intelligent Systems Group,
Department of Computer Science,
Technical University of Cluj-Napoca, Romania
Adrian.Groza@cs.utcluj.ro,nicoleta@isg.cs.utcluj.ro

***Abstract*—Due to the constantly increasing amount of ontologies available, the need for tools supporting ontology evaluation and ranking is crucial. In this paper, we present a Web tool for evaluating and ranking ontologies. The ontologies are evaluated against various structural and semantic metrics. We describe the metrics developed for the ranking system and present the system's results in the context of the Ontology Building Competition 2014 - a learning environment for developing ontologies.**

***Index Terms*—ontology evaluation, ontology metrics, vehicular networks**

I. Introduction

One of the biggest concerns regarding the adoption of ontologies at a wider scale on the web is how to assess and measure the quality of an ontology and which of the ontologies available is most suitable to specific user requirements [4]. This is why ontology evaluation plays an essential role in the process of developing and reusing ontologies.

Even though reuse and sharing of ontologies is recommended, because of the time and costs saving, users encounter difficulties when searching for a relevant ontology. This problem appears because different ontologies can conceptualize the same body of knowledge and we should be able to identify which one best suits some predefined criteria [4]. According to [12], two important limitations must be overcome in order to help users in finding the best ontology for a specific context: the first one refers to the need of an ontology selection mechanism that meets users' requirements and the other one invokes the adjustment of the ontology selection and ranking methods to semantic-match capability.

In this paper, we present two ontology ranking algorithms that were applied for the evaluation of the ontologies uploaded for the BOC 2014 by the students enrolled in the Knowledge Based Systems course.

One of the methods implemented for our ranking system employs a scoring-based aggregation measure, as the one used in the previous edition of the competition [6]. The ranking is computed separately for each criteria and the final score represents the average of the obtained ranks. In order to observe how the user's preferences among criteria influence the final ranking score, a weighted average will also give the user a separated ranking score. The other ranking method evaluates the content of the ontologies to be ranked by querying a set of terms identified by the user in the domain of knowledge. This method was adapted from the AktiveRank [1] and OS_Rank [15] algorithms.

The rest of the paper is organized as follows: The related work is presented in section II. Section III briefly introduces description logic, as the technical instrumentation used to develop ontologies. Section IV details our approach and the system's architecture. In section V, the metrics implemented in our system are described. Section VI states the ranking algorithms applied. Formal specifications of the BOC competition are presented in section VII, followed by the discussion and conclusions in section VIII.

II. Related Work

Even though various approaches of ontology evaluation methodologies are defined, the problem of finding a suitable tool for automatic evaluation and ranking of a certain set of ontologies may affect their usage at a larger scale. According to [4], the evaluation of the ontologies should be done separately at different levels, rather than directly evaluating the ontology as a whole. In this paper, the related work is analyzed from two perspectives: the research in ontology evaluation and in ontology ranking methodologies.

Taking into consideration the first perspective of our study, [12] and [11] define high-level approaches for determining which ontologies are more relevant for a certain purpose and context. The general ontological properties can be assessed by tools like OntoQA [14], OntoClean [7] and OntoMetric [9]. OntoQA proposes a set of schema and structural metrics that evaluate how the classes, relationships and individuals are connected, while OntoClean evaluates the ontologies at the meta-data level, being domain independent. OntoMetric is a framework proposed to measure the suitability of ontologies and OntoRich [3], besides having the capability of calculating some metrics, can also enrich the ontologies from different sources.

Ranking is considered a key element of information retrieval [1] . When ranking ontologies, the chanllenge is take into consideration all their semantic, syntactical, and contextual aspects. The ranking methods from the literature focus rather on evaluating specific parts of ontologies and there is still need of a new ranking algorithm that should take into account more aspects of the evaluation. AktiveRank [1] is a technique for ranking ontologies based on the analysis of concept structures. Similar to this method, the OS_Rank [15] algorithm ranks ontologies based on semantic relations and structure. These two methods use SWOOGLE [5] for searching the ontologies that match some terms from the user queries.

Another type of ranking approach is based on popularity, measured in terms of referrals and number of citations between ontologies. Such a method is defined by the semantic search engines like SWOOGLE [5] and OntoKhoj [13] that use PageRank algorithm to rank ontologies. Due to the fact that ontologies are not so well connected and cited like web pages are, this ranking method may be not so efficient if applied on ontologies.

In order to verify how well the ontologies perform under certain measures, we implemented some of the metrics defined by OntoQA [14] and applied a scoring-based ranking method on the metric scores results. The other ranking approach presented in this paper was inspired from AktiveRank [1] and OS_Rank [15], by taking into consideration some metrics and measures defined in their methodologies. The experiments and detailed specifications of our method are explained in the next sections.

III. Description Logic

In the description logic $\mathcal{ALC}$, concepts are built using the set of constructors formed by negation, conjunction, disjunction, value restriction, and existential restriction [2], as shown in table I. Here, C and D represent concept descriptions, while r is a role name. The semantics are defined based on an interpretation $I = (\Delta^I, \cdot^I)$, where the domain Δ^I of I contains a non-empty set of individuals, and the interpretation function $\cdot^I$ maps each concept name C to a set of individuals $C^I \in \Delta^I$ and each role r to a binary relation $r^I \in \Delta^I \times \Delta^I$. The last column of table I shows the extension of $\cdot^I$ for non-atomic concepts.

TABLE I
Syntax and semantics of $\mathcal{ALC}$.

Constructor	*Syntax*	*Semantics*
negation	$\neg C$	$\Delta^I \setminus C^I$
conjunction	$C \sqcap D$	$C^I \cap D^I$
disjunction	$C \sqcup D$	$C^I \cup D^I$
existential restriction	$\exists r.C$	$\{x \in \Delta^I \vert \exists y : (x,y) \in r^I \wedge y \in C^I\}$
value restriction	$\forall r.C$	$\{x \in \Delta^I \vert \forall y : (x,y) \in r^I \rightarrow y \in C^I\}$
individual assertion	$a : C$	$a \in C^I$
role assertion	$(a,b) : r$	$(a,b) \in r^I$

An ontology consists of terminologies (or TBoxes) and assertions (or ABoxes).

Definition 1. *A terminology $TBox$ is a finite set of terminological axioms of the forms $C \equiv D$ or $C \sqsubseteq D$.*

Example 1 (Terminological box)**.** *In the tbox `Vanet` in fig. 1, the first line specifies the domain of the role $belongsTo$ and its range is defined in the second line. Vehicles are partitioned into private and public (lines 3 and 4). The concept $PublicVehicle$ is further refined in Bus and Police (lines 5 and 6). The axiom 7 specifies that a $PublicVehicle$ should belong only to public agencies. The road side unit used for car-2-infrastructure communication can be deployed both by public or private service providers (line 8).*

1 $\exists belongsTo.\top \sqsubseteq Vehicle$
2 $\top \sqsubseteq \forall belongsTo.(Individual \sqcup Company \sqcup PublicAgency)$
3 $PrivateVehicle \sqsubseteq Vehicle$
4 $PublicVehicle \sqsubseteq Vehicle$
5 $Bus \sqsubseteq PublicVehicle$
6 $Police \sqsubseteq PublicVehicle$
7 $PublicVehicle \sqsubseteq \forall belongsTo.PublicAgency$
8 $RoadSideUnit \sqsubseteq \exists belongsTo.(PublicAgency \sqcup PrivateServiceOp)$

Fig. 1. Modeling VANETs-related knowledge in description logics.

10 (in-abox vanet-cluj Vanet)
11 $b1 : Bus$
12 $lta - brno : LocalTransportAgengy$
13 $rsu1 : RoadSideUnit$
14 $(b1, lta - cluj) : belongsTo$
15 $(rsu1, lta - cluj) : belongsTo$

Fig. 2. Modeling assertions in VANETs.

Definition 2. *An assertional box $ABox$ is a finite set of concept assertions a:C or role assertions (a,b):r, where C designates a concept, r a role, and a and b are two individuals. Usually, the unique name assumption holds within the same $ABox$.*

Example 2 (Assertional box)**.** *The assertional box `vanet-cluj` makes use of the terminologies in the `Vanet` tbox (line 10 in fig. 2). The bus $b1$ (line 11) belongs to the local transportation agency $lta - cluj$ (line 14). Similarly, the road side unit $rsu1$ (line 13) operates under the same public agency $lta - brno$ (line 15).*

A concept C is satisfied if there exists an interpretation I such that $C^I \neq \emptyset$. The concept D subsumes the concept C, represented by $C \sqsubseteq D$, if $C^I \subseteq D^I$ for all interpretations I. Constraints on concepts (i.e. `disjoint`) or on roles (`domain`, `range` of a role, `inverse` roles, `transitive` properties), number constraints (`max`, `min`), role inheritance (`parent` role) can be specified in more expressive description logics[1].

IV. System Architecture

The system architecture is composed of two main modules: the evaluation module and the ranking module, as showed in Fig. 3. The evaluation module contains the metrics implementations that use the OWL API [8] and Pellet Reasoner. OWL API is a high level Application Programming Interface (API) that supports the creation and manipulation of OWL Ontologies. Some key aspects of the OWL API include an axiom-centric abstraction, general purpose reasoner interfaces

[1] We provide only some basic terminologies of description logics in this paper to make it self-contained. For a detailed explanation about families of description logics, the reader is referred to [2].

and support for parsing and serializing ontologies in a variety of syntaxes [8]. Pellet is an OWL-DL reasoner that has extensive support for reasoning with individuals, relationships and user-defined datatypes.

Another important part of the evaluation process is done by the SPARQL support in Jena that is available in our system via a module called ARQ [10]. This module acts as an external SPARQL agent that processes the returning queries' results.

The ranking module consists of the ranking algorithms implementations that use the metrics' evaluation results. The ranking methodologies are described in details in section VI.

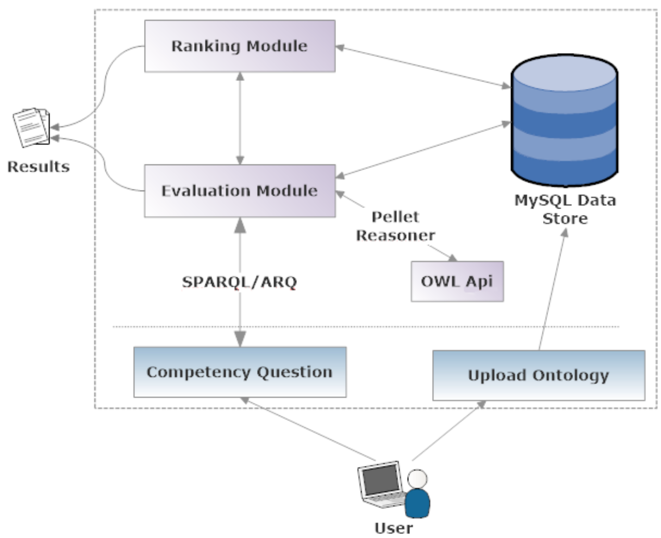

Fig. 3. System's architecture.

The modules presented above are integrated in a Java web-based application, deployed on a Tomcat application server. The ontologies are stored in a relational database after the upload process is done. This way, users can have access to a kind of history of their previous rankings and versions of the ontologies they uploaded.

V. Metrics

The Ontology Evaluation Module provides methods for evaluating the quality of ontologies at two different dimensions: structural and semantic. For the structural evaluation, the metrics implemented indicate how well the ontologies were designed with respect to the size, depth, width, density, richness, inheritance of a schema. The semantic layer of the evaluation is based on instance metrics that check how the instances were placed in the ontology and the usage of the real-world knowledge representation. The following metrics definitions are adapted from [14], [1], [15].

A. Structural Evaluation

The metrics related to the structural layer are also known as schema metrics and they analyze ontologies as graph structures. This kind of evaluation does not verify if the knowledge is correctly modeled in the ontology, but rather how possible it is to represent rich knowledge by using certain ontology design. Some of the metrics developed count the number of the most important elements that enrich an ontology: number of concepts(classes), number of object properties(object relationships), number of data properties (data relationships), number of individuals(instances), number of subclasses. The results of this type of metrics represent the core-base of the further implemented ones.

1) Attribute Richness (AR): Represents the ratio between the total number of attributes(A) in the ontology and the total number of classes(C). Indicates how much knowledge about classes is provided by the ontology on average.

$$AR = \frac{\mid A \mid}{\mid C \mid} \tag{1}$$

2) Relationship Richness (RR): Defined as the ratio between the total number of relationships(R) in the schema and the sum of the number of subclasses(SC) plus the number of relationships(R). Indicates which is the percentage of rich relationships defined from the total number of relationships in the schema (rich and inheritance relationships).

$$RR = \frac{\mid R \mid}{\mid SC \mid + \mid R \mid} \tag{2}$$

3) Inheritance Richness (IR): Represents the ratio between the total number of subclasses(SC) and the total number of classes(C) defined in the schema. This measure aims to identify if the ontology has a vertical or an horizontal structure.

$$IR = \frac{\mid SC \mid}{\mid C \mid} \tag{3}$$

B. Semantic Evaluation

Semantic evaluation checks the amount of the real-world knowledge represented by the ontologies and how well the instances are placed and distributed within the schema. The metrics for this type of evaluation can be grouped in two categories [14]: knowledge base metrics and class metrics. Another part of the semantic evaluation is represented by the competency questions module. It intends to evaluate how well the domain was modeled by interrogating the ontologies with some domain-oriented questions. Depending on the number of answers an ontology provides, we can determine which are the ontologies most representative for certain domains.

1) Knowledgebase metrics:

a) Class Richness (CR): Defined as the ratio of the number of non-empty classes used in the schema(C') to the total number of classes(C) in the ontology. This metric evaluates the distribution of instances across the classes defined, showing the percentage of the data used in the KB schema representation.

$$CR = \frac{\mid C' \mid}{\mid C \mid} \tag{4}$$

b) Average Population (AP): Defined as the total number of instances(I) in the schema divided by the number of classes(C). This metric checks how well the instances were distributed across the classes of the ontology.

$$AP = \frac{\mid I \mid}{\mid C \mid} \tag{5}$$

c) Cohesion (Coh): Represented by the total number of separated connected components(SCC) of the graph in the KB schema. Indicates which are the areas in which instances could be more closely connected.

$$Coh = \mid SCC \mid \tag{6}$$

2) Class metrics: The scores for this type of evaluation are calculated for each class of the ontology. The final result consists in showing the first ten classes having the highest value, the maximum value obtained and how many classes are evaluated as having the score equal to 0.

a) Importance (Imp): Defined as the ratio between the total number of instances that belong to the subtree rooted at the current class($C_i(I)$) to the total number of instances(I) in the schema. This metric result shows the distribution of instances over classes and identifies which are the areas in focus towards the instances extraction process.

$$Imp_i = \frac{\mid C_i(I) \mid}{\mid I \mid} \tag{7}$$

b) Inheritance Richness (IRC): Defined as the number of subclasses in the subtree starting from the current class($SC(C_i)$) to the total number of nodes in the subtree(N).

$$IRC_i = \frac{\mid SC(C_i) \mid}{\mid N \mid} \tag{8}$$

c) Relationship Richness (RRC): Defined as the number of relationships to which the instances of the current class are connected to the total number of relationships defined for the current class. This metric indicates how many properties of a certain class are used at instances level.

$$RRC_i = \frac{\mid R(I_i, I_j) \mid}{\mid R(C_i) \mid}, I_i \in C_i, I_j \in C \tag{9}$$

d) Connectivity (Con): Defined as the number of instances of other classes(I_j) connected to the instances of the class being evaluated($C_i(I)$). This metric indicates which classes play a central role depending on the popularity of instances in the schema.

$$Con_i = \mid I_j, R(I_i, I_j \wedge I_i \in C_i(I)) \mid \tag{10}$$

C. Term-Based Evaluation

1) Class Match Measure(CMM): This measure evaluates the coverage of an ontology with respect to the search terms representing the domain in which the user is interested. It is defined as a weighted sum of the exact matching(EM) and partial matching(PM) of the search terms that correspond to the labels of the classes in the ontology.

Definition 3. *Let $C_n[o] = \{c_1[o], c_2[o], ..., c_n[o]\}$ be the set of classes in ontology o and $T = \{t_1, ..., t_m\}$ be the set of search terms, where n is the total number of classes in the ontology o and m is the total number of terms in the set T.*

Exact matching formula(EM):

$$EM(o,T) = \sum_{i=1}^{n} \sum_{t=1}^{m} E(c_i[o], t_m) \tag{11}$$

where

$$E(c_i[o], t_m) = \begin{cases} 1, & \text{if } label(c_i[o]) = t_m \\ 0, & \text{if } label(c_i[o]) \neq t_m \end{cases}$$

The result of the $EM(o,T)$ represents the total number of classes in ontology o that match exactly some of the terms from the set T.

Partial matching formula(PM):

$$PM(o,T) = \sum_{i=1}^{n} \sum_{t=1}^{m} P(c_i[o], t_m) \tag{12}$$

where

$$P(c_i[o], t_m) = \begin{cases} 1, & \text{if } label(c_i[o]) \text{ contains } t_m \\ 0, & \text{if } label(c_i[o]) \text{ does not contain } t_m \end{cases}$$

The result of the $PM(o,T)$ represents the total number of classes in ontology o that match partially some of the terms from the set T.

Final class match measure formula for ontology o with respect to the set of search terms T:

$$CMM(o,T) = \alpha EM(o,T) + \beta PM(o,T) \tag{13}$$

where α and β represent the exact matching and partial matching weights.

2) Density Measure(DM): Identifies the degree of detail that can be deduced from the knowledge representation of a certain concept. In order to approximate the amount of information content further specified in the schema, the number of subclasses, superclasses, attributes, class importance and class connectivity associated to a matched class are taken into consideration. We added to Alani's [1] original formula two metrics: class importance and class connectivity as we consider they are an important source of finding the degree of detail represented by a certain class of an ontology schema.

Definition 4. *Let $D = \{D_1, D_2, D_3, D_4, D_5\} = \{relations(c), superclasses(c), subclasses(c), importance(c), connectivity(c)\}$ be the set of metrics considered for the density measure.*

$$dem(c) = \sum_{i=1}^{5} w_i \mid D_i \mid \tag{14}$$

$$DEM(o) = \frac{1}{n}\sum_{i=1}^{n} dem(c_i) \quad (15)$$

where w_i is a weight set to a default value of 1 and n represents the total number of exact and partial class matchings, $n = EM(o,T) + PM(o,T)$.

3) Semantic Relationships(SRR): This measure checks the semantic associations of a certain class among the other resources of the ontology. It counts how many of the matched classes are connected to other classes in the ontology through relationships. There are two types of associations: outgoing relationships (eg. if a class c_i is connected to a class c_j) and incoming relationships (eg. if a class c_j is connected to the current class c_i).

Definition 5. *Let $r(c)_{out}$ be the number of outgoing relationships of a class c and $r(c)_{in}$ be the number of incoming relationships of class c.*

$$SR(c) = \frac{r(c)_{out}}{\mid totalr_{out} \mid} r(c)_{out} + \frac{r(c)_{in}}{\mid totalr_{in} \mid} r(c)_{in} \quad (16)$$

where $\mid totalr_{out} \mid$ represents the total number of outgoing relationships of the ontology o and $\mid totalr_{in} \mid$ is the total number of incoming relationships of the ontology o.

The final formula can be calculated as follows:

$$SRR(o) = \alpha \sum_{i=1}^{n} SR(c_i) + \beta \sum_{j=1}^{m} SR(c_j) \quad (17)$$

where n is the number of exact match classes, m is the number of partial match classes and $\alpha > 0, \beta > 0$ are the weights associated to each measure.

VI. Ranking algorithms

In this section, we present the ontology ranking algorithms implemented in our ranking system. The two methods are based on the semantic and syntactical evaluation metrics that were described above.

A. Scoring-Based Ranking Algorithm

The first ranking method used by our system is represented as a linear combination of the metrics scores that are calculated for the set of ontologies. For each metric involved in the ranking process, the ontologies are ranked separately. Thus, each ontology will have a set of ranking scores. The overall ranking for each ontology is computed as the average of the obtained ranks.

Definition 6. *Let $O = \{o_1, o_2, ..., o_n\}$ be the set of ontologies to be ranked, $M = \{m_1, m_2, ..., m_k\}$ the set of defined metrics. The final result is calculated as follows:*

$$Score[o_i] = \sum_{l=1}^{k} \frac{rank_{m[l]_i}}{\mid M \mid}, o_i \in O \quad (18)$$

where $\mid M \mid$ represents the total number of metrics and $rank_{m[l]_i}$ is the rank corresponding to the l-th metric of the i-th ontology.

Another option for this algorithm is to consider weights: for a certain user, a criteria is more important that another one. In this case, the score is computed as follows:

$$Score[o_i] = \sum_{l=1}^{k} \frac{rank_{m[l]_i} * w_{m[l]}}{\mid M \mid}, o_i \in O \quad (19)$$

where $\mid M \mid$ represents the total number of metrics and $rank_{m[l]_i}$ is the rank corresponding to the l-th metric of the i-th ontology and $w_{m[l]}$ is the weight associated to that metric, where $\sum_{i=1}^{l} w_i = 1$.

For each criteria, the obtained metric scores are sorted in descending order and the rank is given by the position in the sorted list. There are some metrics (eg. number of violations, number of unsatisfiable classes, number of not connected classes) that are sorted in ascending order and the rank is given by their corresponding position in the list. The metrics that are taken into account by the ranking algorithm are the ones defined in section IV.

B. Term-Based Ranking Algorithm

This algorithm is adapted from AktiveRank [1] and OS_Rank [15] algorithms, described in section II. The following measures are computed before applying the algorithm: class match measure (CMM), density (DM) and semantic relations ranking (SRR).

The final score for each ontology can be obtained after applying the three measures (CMM,DM,SRR) to all the ontologies to be ranked. The measures are normalized in the range [0,1] by dividing each of them with the maximum value obtained for each type of measure applied. The overall ranking score is calculated by aggregating the normalized measures' values, considering a weight for each type of measure. This way, the final ranking value can be adjusted by changing the weights according to the importance the user gives to each of them.

Definition 7. *Let $O = \{o_1, o_2, ..., o_n\}$ be the set of ontologies to be ranked, $M = \{m_1, m_2, m_3\}$ = $\{CMM, DM, SR\}$ the set of measures and w_i a weight factor, corresponding to each type of measure, where $w_i > 0$ and $w_1 + w_2 + w_3 = 1$.*

$$Score[o_i] = \sum_{i=1}^{3} w_i \frac{m[i]}{maxM_i}, o_i \in O \quad (20)$$

where $maxM_i$ is the maximum score obtained for the i-th metric type.

As in OS_Rank, a weight can also be set to each term in the query. The formula in this case is defined as:

Definition 8. *Let $O = \{o_1, o_2, ..., o_n\}$ be the set of ontologies to be ranked, $Score[o_{it}]$ is the score corresponding to the*

1.$Team \sqsubseteq (< 3) hasMember.Student \sqcap \exists hasDeveloped.Ontology$
2.$Competition \sqsubseteq hasTrack.Track \sqcap (> 1) hasTeam.Team$
3.$Competitor \equiv Student \sqcap \exists hasTask.CreateOntology$
4.$Organiser \equiv Student \sqcap hasTask.DefineMetrics \sqcap hasTask.DefineRankingMethods$
5.$StructuralMetric \sqsubseteq Metric$
6.$SemanticMetric \sqsubseteq Metric$
7.$SizeMetric \sqsubseteq StructuralMetric$
8.$KnowledgeBaseMetric \sqsubseteq SemanticMetric$
9.$ClassMetric \sqsubseteq SemanticMetric$
10.$ComepetencyQuestion \sqsubseteq SemanticMetric$
11.$RankingScore \equiv Score \sqcap \exists hasDefined.Metric \sqcap \exists hasDefined.RankingMethod$
12.$FinalScore \equiv Score \sqcap \exists hasCalculated.RankingScore$

Fig. 4. Formal rules of the competition in Description Logic.

i-th ontology with the t-th term and w_t a weight factor, corresponding to each term in the query, where $w_t > 0$, $\sum_{t=1}^{n} w_t = 1$ and n represents the total number of terms in the query.

$$WeightedScore[o_{it}] = \sum_{t=1}^{n} w_t * Score[o_{it}], o_i \in O \quad (21)$$

VII. Validation against BOC 2014

The Ontology Building Competition represents a new learning environment which allows students to manage ontologies and to assess the quality of their building process automatically. The competitors are students enrolled in the Knowledge Based Systems course and had the task of developing ontologies for the tourism and car vanets domains.

A. BOC Meta-Ontology

The competition is formally described as a meta-ontology in which the rules of the competition, the competitors, their scores at each criteria and the final rankings are defined. The reference ontology is enriched automatically each time a new team register into the competition, when new metrics, tracks or competency questions are defined, but also when the metrics and rankings are calculated. This way, the meta-ontology can be extended easily and the use of a relational database is not needed anymore. A formal specification of the competition's meta-ontology is presented in Description Logic in Fig. 4.

Based on the concepts introduced above, the competition can be parameterized in terms of type of metrics, ontology domains and ranking algorithms (see Fig. 5). Hence, the individual $c2014$ is a competition which has two domains: tourism, respectively vehicular networks and it took place in 2014.

B. Competency Questions

A set of 10 questions were defined in natural language for each track of the competition. The competitors needed to adapt their ontologies in order to cover the information required to provide answers to these questions. The queries tried to evaluate how well the domain was modeled with respect to the the implicated actors (eg. tourists, drivers). Hence, in case of the tourism domain, some of the questions regarded type of accommodations, type of services, activities and points of interests, as for the vanets domain, the questions contained key terms like radars, accidents, traffic. For example, an ontology for the tourism domain should be able to provide answers to the questions presented in Fig. 6.

In case of the vehicular networks track, some questions defined are presented in Fig. 7.

$c2014 : Competition$
$tourism : Track$
$vanets : Track$
$(c2014, tourism) : has - track$
$(c2014, vanets) : has - track$
$(c2014, 2014) : holds - in$
$inheritanceRichness : SizeMetric$
$numberOfIndividuals : SizeMetric$
$classImportance : SemanticMetric$
$classConnectivity : SemanticMetric$
$maria : Competitor$
$scoringBasedAlg : RankingAlgorithm$

Fig. 5. Instantiating the competition.

"What restaurants are there within a given distance?"

```
 SELECT ?name WHERE {
    ?restaurant dbtourism:name ?name
    ?restaurant dbtourism:withinDistance ?distance
FILTER(?distance < 10) }
```

"Which are the types of accommodation defined in the ontology?"

```
 SELECT ?accommodationType WHERE {
    ?accommodation dbtourism:isType
?accommodationType}
```

Fig. 6. SPARQL queries for tourism-related competency questions.

"Is there a radar on street X?"

```
ASK {
    ?street dbvanets:name X
    ?radar dbvanets:isOnStreet ?street }
```

"Is there an accident on street X?"

```
ASK {
    ?street dbvanets:name X
    ?accident dbvanets:isAccidentOnStreet ?street }
```

Fig. 7. SPARQL queries in the vehicular networks domain.

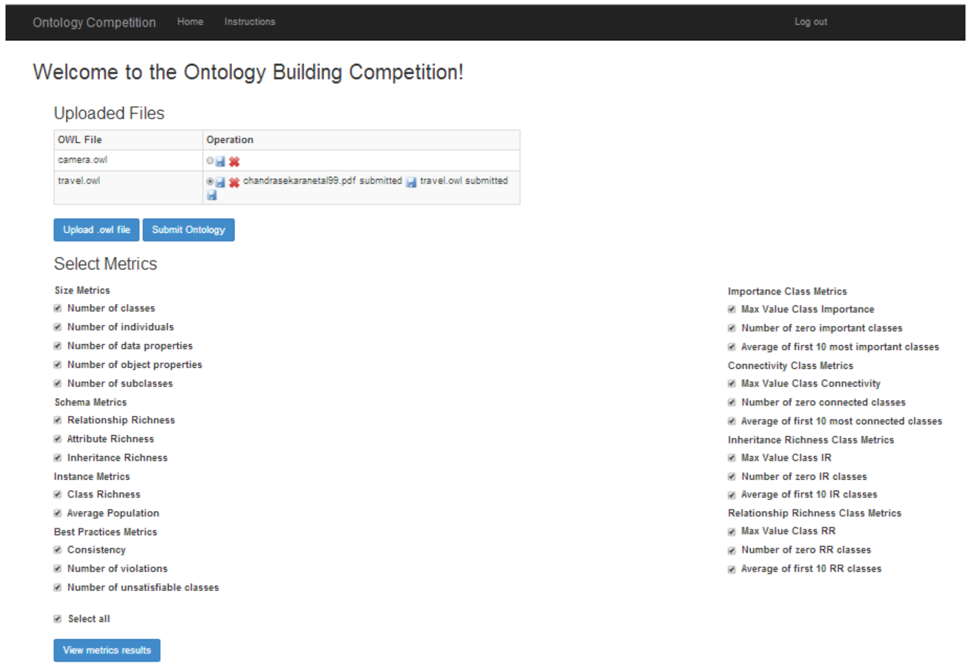

Fig. 8. Selecting active metrics for ontology ranking.

C. BOC System Functionalities

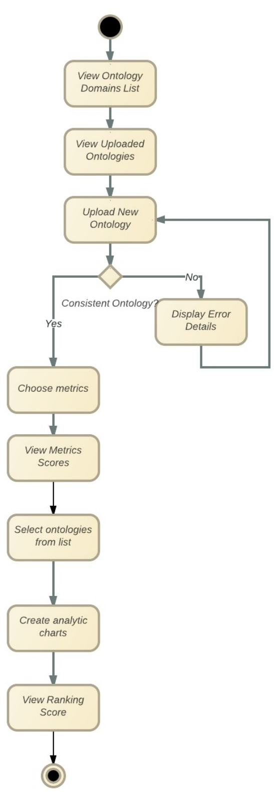

Fig. 9. System's workflow.

The web ranking system proposed is used for the management and ranking of the ontologies uploaded to the BOC competition. It can also act as an evaluation tool for any user eager to compare the quality of some ontologies. Users can upload their ontologies in the system, view the metrics results and then choose the ontologies to be ranked according to the selected criteria. This functionality is important when: i)users want to see the evolution of the ontologies they build; ii)they want to verify how their ontologies quality is influenced by the import of another ontology. This way, a history of the previous rankings done can be kept and viewed by the user anytime. Another important characteristic of our system is that it allows users to adapt the ranking algorithms they want to apply according to their needs. They can select which of the available metrics to be taken into consideration by the ranking algorithm. Furthermore, they can set a weight to each metric according to its relevance and importance in the final ranking result. Figure 8 shows the main page of the system and figure 9 presents a diagram with the main system's workflow.

D. Experiments and Results

In order to validate the correctness of the results generated by our system, we have manually evaluated the ontologies uploaded to the BOC competition. Then, we set a rank to each of them and compared the results with the ones given by the system.

There was a number of 20 teams that registered and built ontologies for two domains: tourism and vehicular networks. The manual evaluation was done with respect to the structural and semantic associations of the elements of the ontologies.

In Table II the average of the obtained metrics results involved in the ranking system is showed. The uploaded ontologies were first tested with respect to their consistency. All of them proved to be consistent and could be further

TABLE II
METRICS IN BOC2014.

Number of Classes	79
Number of Individuals	380
Number of Data Properties	9.31
Number of Object Properties	18.4
Number of Subclasses	86.04
Relationship Richness	0.33
Attribute Richness	0.57
Inheritance Richness	1.05
Class Richness	0.4
Average Population	6.83
Maximum Value Importance	0.32
Maximum Value Connectivity	5.4
Maximum Value Relationship Richness	0.45

evaluated. From the structural point of view, the ontologies had between 7 and 238 classes, so an average of 79 classes. The number of individuals showed big difference from an ontology to another, as there were some ontologies with only 9 individuals, while one of them had 4924 individuals and an average of 380. The ontologies were not so rich with respect to the data and object properties that varied from 2 to 47.

As for the semantic evaluation, we computed the maximum class importance, class connectivity and relationship richness from each ontology, as well the number of classes in each ontology that are not connected at all with the instances of other classes. The values obtained showed that even though some of the metrics had a lot of individuals, they were not so well connected to the other resources of the ontologies. This is why the values obtained in this part of the evaluation are not so high, which means that the ontologies could be further improved with respect to their semantic quality.

VIII. DISCUSSION AND CONCLUSIONS

An important aspect of our evaluation system is that it is not domain-dependent. Users have no restriction when evaluating their ontologies besides uploading an owl file.

The system can also act as a repository for ontologies, in which users can download and extend the ontologies uploaded by others, if they are public.

An improvement of this system would be to integrate it with an external ontology search engine, like SWOOGLE. REST services of SWOOGLE can be easily added to our system. This way, users could compare the quality of the ontologies they built with the ones retrieved from SWOOGLE. The process of finding a proper ontology for a certain purpose could have great impact on the performance of web applications.

One of the objectives of this study was to determine and analyze the needs and attitude of students towards working with ontologies, as a “knowledge-based infrastructure that supports the representation and sharing of domain knowledge” [12]. The role of ontologies in future education is becoming more and more important as the development of technology brings with it the need of new skills for interpreting and integrating the constantly growing amount of information that can be found on the Internet.

As conclusions, we analyzed and identified some methods for the evaluation and ranking of the ontologies. The usefulness of our system was validated in the context of the building competition. We took into consideration different types of metrics and we evaluated the quality of the ontologies from both syntactical and semantic point of view. Even though it is difficult to assess the quality of ontologies in terms of semantic capabilities, we focused rather on evaluating semantic associations between the elements of ontologies. Our ranking system aims to assist the user in the process of ontology engineering by providing relevant and quantitative results regarding the quality of the ontologies. This way, knowledge information can be managed and extended easier by using ontologies.

ACKNOWLEDGMENTS

We thank the reviewers for their valuable comments. This research was supported by the Technical University of Cluj-Napoca, Romania, through the internal research project ”GREEN-VANETS: Improving transportation using Car-2-X communication and multi-agent systems”.

REFERENCES

[1] H. Alani, C. Brewster, and N. Shadbolt, “Ranking ontologies with AKTiveRank,” in *In Proc. of the International Semantic Web Conference, ISWC*. Springer-Verlag, 2006, pp. 5–9.

[2] F. Baader, *The description logic handbook: theory, implementation, and applications*. Cambridge university press, 2003.

[3] G. Barbur, B. Blaga, and A. Groza, “OntoRich-a support tool for semi-automatic ontology enrichment and evaluation,” in *Intelligent Computer Communication and Processing (ICCP), 2011 IEEE International Conference on*. IEEE, 2011, pp. 129–132.

[4] J. Brank, M. Grobelnik, and D. Mladenić, “A survey of ontology evaluation techniques,” 2005.

[5] L. Ding, T. Finin, A. Joshi, R. Pan, R. S. Cost, Y. Peng, P. Reddivari, V. Doshi, and J. Sachs, “Swoogle: a search and metadata engine for the semantic web,” in *Proceedings of the thirteenth ACM international conference on Information and knowledge management*. ACM, 2004, pp. 652–659.

[6] A. Groza, B. Varga, and V. Mario, “A learning environment for building and evaluating ontologies: Case study of 2013 Ontology Building Competition,” in *The 10th eLearning and Software for Education Conference - eLSE 2014*, 2014, p. to appear.

[7] N. Guarino and C. Welty, “Evaluating ontological decisions with OntoClean,” *Communications of the ACM*, vol. 45, no. 2, pp. 61–65, 2002.

[8] M. Horridge and S. Bechhofer, “The owl api: A java api for owl ontologies,” *Semantic Web*, vol. 2, no. 1, pp. 11–21, 2011.

[9] A. Lozano-Tello and A. Gómez-Pérez, “Ontometric: A method to choose the appropriate ontology,” *Journal of Database Management*, vol. 2, no. 15, pp. 1–18, 2004.

[10] P. McCarthy, “Search RDF data with SPARQL: SPARQL and the Jena Toolkit open up the Semantic Web, in developerworks,” 2005.

[11] S. Oh and H. Y. Yeom, “A comprehensive framework for the evaluation of ontology modularization,” *Expert Systems with Applications*, vol. 39, no. 10, pp. 8547–8556, 2012.

[12] J. Park, S. Oh, and J. Ahn, “Ontology selection ranking model for knowledge reuse,” *Expert Systems with Applications*, vol. 38, no. 5, pp. 5133–5144, 2011.

[13] C. Patel, K. Supekar, Y. Lee, and E. Park, “OntoKhoj: a semantic web portal for ontology searching, ranking and classification,” in *Proceedings of the 5th ACM international workshop on Web information and data management*. ACM, 2003, pp. 58–61.

[14] S. Tartir, I. B. Arpinar, M. Moore, A. P. Sheth, and B. Aleman-meza, “OntoQA: Metric-based ontology quality analysis,” in *IEEE Workshop on Knowledge Acquisition from Distributed, Autonomous, Semantically Heterogeneous Data and Knowledge Sources*, 2005.

[15] W. Yu, Q. Li, J. Chen, and J. Cao, “OS_RANK: Structure analysis for ontology ranking,” in *Data Engineering Workshop, 2007 IEEE 23rd International Conference on*. IEEE, 2007, pp. 339–346.