An Ontology Selection and Ranking System Based on the Analytic Hierarchy Process

Adrian Groza*, Irina Dragoste*, Iulia Sincai*, Ioana Jimborean*, Vasile Moraru†,
*Department of Computer Science, Technical University of Cluj-Napoca
Adrian.Groza@cs.utcluj.ro
†Department of Applied Informatics, Technical University of Moldova
moraru@mail.utm.md

***Abstract*—Selecting the desired ontology from a collection of available ones is essential for ontology reuse. We address the problem of evaluating, ranking and selecting ontologies according to user preferences. We exploit the Analytic Hierarchy Process (AHP) to solve the multiple-criteria decision problem and to model the preferences of the users. We use AHP to analyze the available ontologies from different perspectives and at different abstraction levels. The decision is based on the concrete end-node measurements and their relative importance at higher levels. For supporting the selection decision, we developed an ontology representation, reasoning and management system. The system applies different metrics on ontologies in order to feed the Analytic Hierarchy Process with facts. The running scenario applies our method to the task of reusing ontologies from the tourism domain.**

***Index Terms*—Ontology selection, ontology ranking, analytic hierarchy process, ontology metrics**

I. Introduction

The task of selecting the most adequate ontology from available repositories is a multi-criteria decision problem. The decision problem can be viewed from various perspectives of multiple stakeholders, each having different goals, which makes it difficult to formulate, understanding and select the most adequate ontology [23].

There are three main strategies to evaluate ontologies: (1) by comparing the ontology with a benchmark ontology (*goal standard evaluation*), (2) assessing the competency of the ontology to complete a task (*task-based evaluation*) and (3) considering various criteria such as consistency, completeness, conciseness (*criteria-based evaluation*) [26]. In line with criteria-based evaluation, we apply the analytic hierarchy process method in the evaluation, ranking and selection of ontologies for knowledge reuse.

The analytic hierarchy process (AHP) [24], [5] is a widely-used method for multicriteria decision support. The method hierarchically decomposes the objectives and evaluates preferences through pairwise comparisons to select among alternatives with multiple attributes. Its most important strengths are [18]: i) simplicity, ii) applicability to group decisions, iii) the possibility to compare quantitative criteria with qualitative criteria, and iv) the allowance of small inconsistency that better accommodates human thinking.

The aim is to support complex and justifiable decisions when selecting ontologies for knowledge reuse. Our AHP-based solution helps ontology users analyze the impact of each ontology attribute in a system or use context and formulate arguments for an informed, rational, complex decision. It also helps users define the relative importance each attribute should have through pairwise comparisons that allow inconsistencies and model human thinking.

The rest of the paper is organised as follows: Section II briefly introduces the AHP method. Section III shows how we apply the AHP method to the ontology evaluation task. Section IV details the architecture of the ontology evaluation system. Section V illustrates the method when selecting ontologies in the tourism domain. Section VI discusses related work, while section VII concludes the paper.

II. Analytic Hierarchy Process for Complex Decision

The Analytic Hierarchy Process (AHP) is a multi-criteria decision-supporting method. AHP structures the complex decision elements in a hierarchical abstraction consisting of: (1) a goal, (2) abstract high-level criteria and their lower levels sub-criteria, and (3) alternatives. The user preferences are modeled using a pairwise comparison matrix [19].

Definition 1. *The Pairwise Comparisons (PC) matrix is a positive reciprocal matrix ($a_{ij} = a_{ji}^{-1}, \forall i, j$) in which the intersection of a line and column represents the relative "weight" of the line criterion in comparison with the column criterion.*

Each non-leaf node from the criteria tree is associated a PC matrix, which contains judgments on the pairwise "weighting" of its children nodes expressed by the decision maker. The user analyzes and compares the importance, preference or impact of each criterion in the matrix in comparison with another: $a_{ij} = a_i/a_j$

Choosing pairwise comparisons as a model to represent preference between criteria has as benefits simplicity: the user does not need to compute a global weight, but focuses on a small, concrete part of the problem and comparing only two elements at a time. Hence, the AHP method allows inconsistency in preference. Criteria is analyzed at multiple abstraction levels and from multiple perspectives, making it suitable for group decision making.

Definition 2. *A reciprocal matrix A is said to be cardinally consistent if $a_{ij} = a_{ik}a_{kj}$ $\forall$ i,j,k where a_{ij} is called a* direct judgment, *given by the decision maker, and* $a_{ik}a_{kj}$ *is an* indirect judgment.

Definition 3. *A reciprocal matrix A is said to be ordinarily transitive (ordinally consistent) if* $\forall i \quad \exists j,k$ *s.t.* $a_{ij} \geq a_{ik} \Rightarrow a_{jk} \leq 1$.

Inconsistency is part of human natural thinking - integrating it in a system can increase its usability. Inconsistent comparisons tend to "weight each-other out", finally expressing user true preferences. Errors introduced by inconsistency can be measured and controlled by the decision aiding system. For example, the system can allow a limited degree of inconsistency, asking the user to re-evaluate his preferences.

PC matrices are used for elicitation of *normalized overall values* for criteria preferences, called *weights*. Thus, from a PC matrix modeling relative preferences for all sub-criteria of a criterion node *k*, we can deduce the weights of each sub-criteria of *k*. The overall weight of a criteria is calculated in the eigenvector [19] of the PC matrix between brother nodes, which is used to compute the global value of each alternative as a weighted sum.

For each non-leaf criterion, a right eigenvector $w = (w_1, ..., w_n)$ is calculated from its *n*-size PC matrix, using the equation:

$$Aw = \lambda_{max} w \tag{1}$$

where *A* is the PC matrix, λ_{max} is largest eigenvalue of *A* that can be calculated from its characteristic equation:

$$det(A \setminus \lambda I) = 0 \tag{2}$$

I being the identity matrix and λ_{max} being the only real and positive solution.

The eigenvector, also called preference vector, holds the overall normalized weight of each sub-criterion represented in the PC matrix. It is real and positive and unique with respect to a multiplicative constant. w_i represents an estimate of the overall weight of criterion *i*[1].

III. Applying Analytic Hierarchy Process for Ontology Selection

To apply AHP method to ontology evaluation, we need two steps: Firstly, we define criteria tree used for ontology selection. Secondly, we define the pairwise comparison matrix used to rank the available ontologies.

A. AHP for Ontology Evaluation

When evaluating an ontology, one must consider its relevance to given domain, but also the quality of the ontology in terms of language expressiveness, size, cohesion, complexity, consistency. These abstract characteristics can be defined with the help of quantitative or qualitative sub-characteristics. Their relative importance depends on the context in which the selected ontology will be re(used) and on the user needs [15].

[1]We provide only some basic terminologies of the AHP in this paper to make it self-contained. For a detailed explanation, the reader is referred to [19], [18].

The candidate ontologies to be evaluated are selected from various repositories based on their coverage of the given domain. Our method for evaluating domain coverage is inspired from OntologyRank [16] and it is based on semantic similarity between terms. We propose an automated improvement of similar evaluation frameworks like OntoMetric [12]. The user needs to specify: (i) the domain description and (ii) pairwise comparisons of the presented criteria importance. The system outputs a ranking of the available ontologies and an evaluation report that explains the decision of the system.

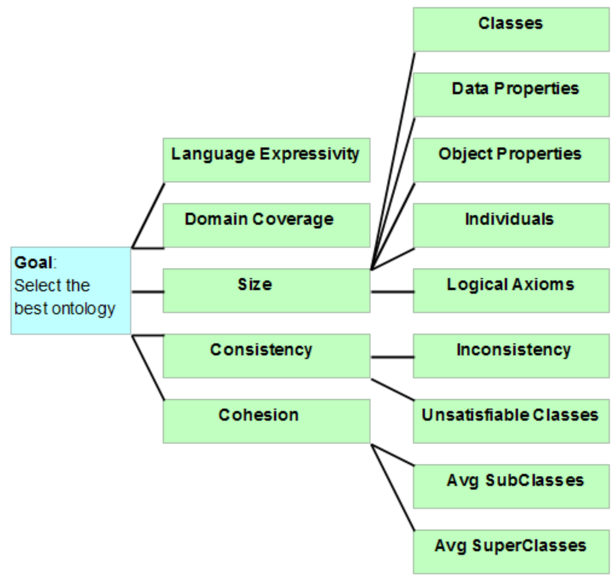

Fig. 1. Criteria tree used for AHP-based ontology selection.

The hierarchical structure proposed by AHP as an abstract model for the decision problem involves a thorough analysis of the problem domain. The proposed model presented in Fig. 1 was inspired from the criteria used in [12] and [14]. We grouped the selected features in five concern classes:

- *domain coverage*: expresses to what degree classes, individuals, and relations specified by the decision maker are found in an ontology for a given domain;
- *size* of the ontology: counts the number of classes, data and object properties, individual, logical axioms, etc.;
- *consistency*: measures whether the ontology has inconsistencies or not, as well as unsatisfiable or cycled concepts;
- *cohesion*: describes the general shape of the ontology taxonomies (the average number of superclasses and subclasses of each class). For instance, in our view, small modules with high cohesion are preferred for performance reasons [14], modularization tools having been developed for this purpose. Cohesion can also describe the quality of a taxonomy [12].
- *language expressivity*, important for representing knowledge in more expressive formal lannguage.

Note in Fig. 1 that the decision goal and the criteria tree are independent from the decision alternatives (the ontologies to be evaluated). The evaluation criteria can be classified as: *qualitative* (i.e., language expressivity, inconsistency) and *quantitative* (i.e., number of classes, individuals, unsatisfiable concepts, average number of sub-classes, domain coverage). Part of the criteria are *positive*, for which a large value is desired at evaluation (i.e., size, domain coverage), while the remaining part are *negative* (i.e., number of unsatisfiable concepts, inconsistency). The decision is based on the values measured in the leaf-nodes of the criteria tree and their relative importance at more abstract levels.

B. Evaluating Alternatives: Pairwise Comparisons Matrix, Eigenvector Method, Additive Normalization Method

Many AHP implementations use pairwise comparisons matrices to evaluate alternatives and obtain normalized values at atomic criteria level ([12], [20]). Our proposed solution uses automated ontology measurements for alternative evaluation.

Because *zero values* may appear often in the proposed ontology measurements, all methods based on the PC matrix (Eigen Vector, Geometric Mean etc) are not applicable for normalizing alternatives leaf-node values, considering the division by zero situation.

We adapt the Additive Normalization (AN) method [25] to *evaluate alternatives for leaf-node criteria.*

Definition 4. *Weighted Arithmetic Mean normalized measurement of alternative i against leaf node criterion :*

$$\overline{leaf_i} = \frac{leaf_i}{\sum_j leaf_j}$$

Alternative i value for leaf criterion $V_i leaf$:

$$V_i leaf = \begin{cases} \overline{leaf_i}, & \text{if criterion leaf is positive and} \\ 1 - \overline{leaf_i}, & \text{if criterion leaf is negative} \end{cases}$$

In order to obtain the sum of normalized values equal to 1, only in the case of negative criteria, a post normalization step is performed:

$$V_i leaf = V_i leaf / \sum_j V_j leaf \quad \text{if criterion leaf is negative}$$

An alternative to *Weighted Arithmetic Mean* is to normalize by the maximum alternative value, instead of the sum defined in 4. This method would obtain a greater difference in normalized alternative values, increasing the impact of each alternative value. It would also preserve the ratio between alternatives better than *Weighted Arithmetic Mean*

This method is also suggested in [10], referencing the *rank reversal* problem solution proposed by Belton and Gear [3], called *Ideal Mode AHP*. Rank reversal may occur when adding a new alternative very similar to an existing one, or deleting one from the previous set. It can be prevented by dividing each value in the rank vector with the maximum value, obtaining values less or equal to 1 that do not sum to 1.

Definition 5. *Maximum normalized measurement of alternative i against leaf node criterion :*

$$\overline{leaf_i} = \frac{leaf_i}{max(leaf_j)}$$

TABLE I
METHODS FOR EVALUATING ALTERNATIVES

Elicitation Method	Mandatory steps	Sum to 1
Weighted Arithmetic Mean	**step 1:** $\overline{leaf_i} = leaf_i / \sum_j leaf_j$ **step 2:** $V_i leaf = \begin{cases} \overline{leaf_i}, & \text{if leaf} \geq 0 \\ 1 - \overline{leaf_i}, & \text{if leaf} < 0 \end{cases}$	**step 3:** $V_i leaf = \frac{V_i leaf}{\sum_j V_j leaf}$ if leaf < 0
Max Normalization	**step 1:** $\overline{leaf_i} = leaf_i / max(leaf_j)$ **step 2:** $V_i leaf = \begin{cases} \overline{leaf_i}, & \text{if leaf} \geq 0 \\ 1 - \overline{leaf_i}, & \text{if leaf} < 0 \end{cases}$	**step 3:** $V_i leaf = \frac{V_i leaf}{\sum_j V_j leaf}$

A consequence of this method is that the sum of normalized alternative values for a criterion would no longer be 1, like the elicited weights of criteria obtained from Eigenvalue Method. The post-normalization step can be performed both for positive and for negative criteria, which would generate identical results to the Weighted Arithmetic Mean for positive criteria, but a greater difference in values for negative criteria. Hence, the possibilities of eliciting alternatives against atomic (leaf) criteria without using *PC matrices*, thus including zero measurement values and negative (cost) atomic criteria, are the following:

Algorithm 1 lists the steps for computing the *global ranking value* V_i for an alternative ontology *i*. Algorithm 2 lists the steps performed during AHP-driven ontology evaluation. Given a set of available ontologies, the set of keywords that define the domain and the criteria tree used by the AHP method, the algorithm outputs the evaluation values of each ontology. In the first step the user updates the domain by including part of the synsets provided by the Wordnet dictionary. Then, for each available ontology, the domain coverage is calculated against the set of words $\mathcal{W}_D$ for the domain D. The ontologies satisfying the threshold δ for domain coverage are selected for possible reuse. Only these ontologies in the set $\mathcal{O}_\delta$ will be analysed with the AHP method.

IV. SYSTEM ARCHITECTURE

The AHP Ontology Evaluation System consists of multiple modules with specific functionality: *Domain Coverage Module*, *Ontology Measurements Module* and *AHP Module*, as shown in Fig. 2. The architecture allows the modules to be reused individually for a desired functionality[2].

A. Domain Coverage Module

This module is used to determine the degree to which available ontologies from a repository are semantically relevant to a certain knowledge domain. The user has the option to preselect ontology alternatives with a domain coverage higher then a *flexible threshold* for the AHP evaluation mechanism. Note that the domain coverage of an alternative, measured

[2]The system is available at http://cs-gw.utcluj.ro/∼adrian/tools/ahp

Algorithm 1: Steps for computing ranking.

Input: $\mathcal{T}$ - the criteria tree
Output: V_i, the global ranking value of the ontology i

foreach *non-leaf criterion* $k_{nleaf} \in \mathcal{T}$ **do**
    construct A_k PC matrix and calculate $\lambda_{max} k_{nleaf}$
foreach *leaf-node criteria* $k_{leaf} \in \mathcal{T}$ **do**
    measure all ontology alternatives against k_{leaf}
    apply AN to calculate $V_i k_{leaf}$
foreach *non-leaf criterion* $k_{nleaf} \in \mathcal{T}$ **do**
    calculate $V_{ik} = V_{i1} * w_{1k} + V_{i2} * w_{2k} + ...$, from bottom levels to top levels,
    where $(w_{1k}, *w_{2k}, ...) = \lambda_{max} k$ of non-leaf criterion k_{nleaf} and V_{ik} represents the value of alternative i evaluated against criterion k_{nleaf}.
    $V_i root = V_i goal = V_i$, the global ranking value of the ontology i.
return V_i

Algorithm 2: Required steps for AHP-based ontology evaluation.

Input: $\mathcal{O}$ - set of candidate ontologies; $\mathcal{W}$ - set of keywords of the domain; $\mathcal{T}$ - the criteria tree
Output: $\langle ontology, [evaluation\ values] \rangle$, association list with ontologies as keys and evaluation values as pairs

$\mathcal{W_D} \leftarrow Update(\mathcal{W}, Wordnet)$
foreach $o \in \mathcal{O}$ **do**
    $DomainCoverage(o, \mathcal{W}_D)$
Define a *domain coverage threshold* $\delta \geq 0$
$\mathcal{O}_\delta \leftarrow Select(\mathcal{O}, \delta)$
foreach $o \in \mathcal{O}_\delta$ **do**
    foreach *non-leaf criterion* $k \in \mathcal{T}$ **do**
        complete the PC matrix to determine the weights of its sub-criteria
    foreach *leaf (atomic) criterion* $k \in \mathcal{T}$ **do**
        $OntologyMetrics(k, o)$
    Normalize ontology measurements to obtain weights for alternatives
    $WeightedSum(o, PC)$

inside this module, is also considered as a *high-level criterion* in the AHP criteria tree (recall Fig. 1). In line with [16], our solution for determining the domain coverage of an ontology involves both *lexical and semantic search* of desired terms in the available ontologies' content. Semantic search is assured by: i) the use of *synonyms* and ii) *polysemy disambiguation*.

The desired knowledge domain is described by a set of user input search terms, that have a specific structure: a term is composed of a main word (noun) and a list of synonyms of that word. Both the main noun and its synonyms are then searched for in the set of concepts of available ontologies.

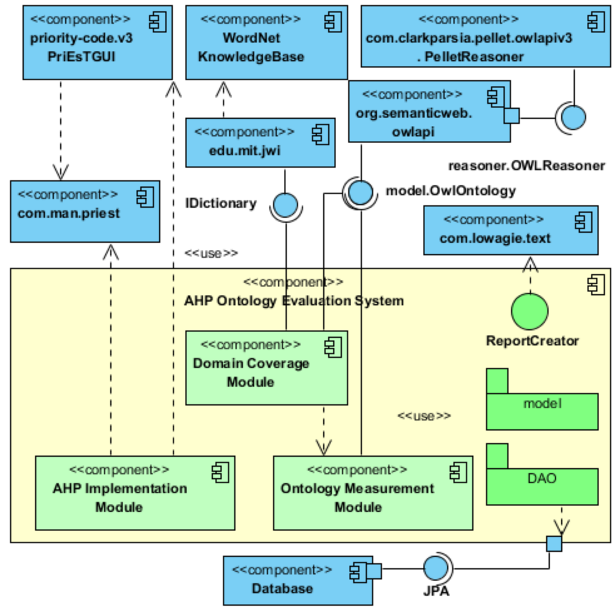

Fig. 2. System architecture.

This study proposes a simplified solution for measuring the domain coverage of an ontology, by counting the number of user-given terms matched in the ontology. A term is *matched* if it or one of it's *synonyms* was found as a class defined in the ontology. The user needs to provide only a set of nouns that define concepts and their synonyms.

Definition 6. *Given a pair* $\langle t_i, Syn(t_i) \rangle$*, where term* t_i *is a* noun *describing a concept and* $Syn(t_i)$ *is a (possibly empty) set of* synonyms *of term* t_i*, one can define a certain domain or* semantic field *trough a set of terms and their synonyms:*

$$T = \{ \langle t_i, Syn(t_i) \rangle \quad | i \geqslant 1 \}$$

Definition 7. *The coverage of a given domain* T *for an ontology* O *is the ratio of terms matched by classes of the ontology:*

$$DomainCoverage(T, O) = \frac{matched(T, O)}{|T|},$$

where $|T|$ *counts the* $\langle t_i, Syn(t_i) \rangle$ *pairs;* $matched(T, O) =$ *the number of pairs* $\langle t_i, Syn(t_i) \rangle$ *for which* $\exists$ *a class* $c \in O$ *s.t.* $c = t_i$ *or* $c \in Syn(t_i)$

Synonyms are obtained dynamically from *WordNet*, containing sets of cognitive synonyms called *sysnets*, each related to a different concept. The *sysnets* form a network of lexically and semantically connected words. A list of synonyms for every user given word is displayed, for supporting the user to select the most suitable synonyms.

Polysemy disambiguation is eventually assured by the user. An aspect that helps the user disambiguate is displaying the list of synonyms for all senses of the given word, from which users can replace the main word with a more specific synonym. Polysemy is also avoided by grouping synonyms by term meaning and displaying a definition of each word semantic. This helps the user select only synonyms that belong to the desired sense.

B. Ontology Metrics Module

The evaluation module includes various metrics for evaluation ontologies proposed in literature [22], [1], [27]. We focus here on the evaluation criteria specified by the criteria tree of the AHP methodology (recall Fig. 1). Thereby, we describe the metrics regarding size, consitency, cohesion, domain coverage and language expressivity. The first four criteria are detailed in Table II, while the language expressivity refers to the family of description logic used to represent knowledge: the more expressive the language, the better the ontology, given no explicit requirements on the complexity of reasoning.

Size metrics refer to general properties that enrich an ontology. They are important for the analysis of the ontologies' structural richness. These metrics include: number of concepts, number of relationships, number of individuals, or number of rules. Size metrics represent the core-base of the more complex metrics.

Once all the ontologies were evaluated, the tool generates various reports with i) a detailed view of all metrics used (their name, category, definition, explanation) and ii) a description of the ontology (its id, URI, last measurements update date) and the values obtained in each measurement. The report can be used for the following purposes:

- *information*: to learn about certain metrics that can be applied to ontologies;
- *ontology content analysis*: to obtain a wide range measurements about ontologies automatically;
- *AHP*: to understand the AHP criteria established in our system in order to express preferences between them;
- *document*: to document and justify decisions regarding ontology reuse/implementation, which were made rationally, based on obtained data.

For the consistency-related metrics, the system counts number of unsatisfiable classes and the number of inconsistencies.

Cohesion (Coh) represents the total number of separated connected components (SCC) of the ontology graph: $Coh = \mid SCC \mid$. The metrics indicates which are the areas in which instances could be more closely connected.

C. AHP module

The Analytic Hierarchy Process was implemented in our project using *PriEst Tool* [20], adapted to the project scope. *PriEst* is a priority estimation research project, which provides the domain model for AHP, various elicitation methods, consistency and accuracy evaluation. It provides also a visual representation of the model and decision aiding features for preference judgments.

The capabilities of the decision maker are augmented with the following: (i) fill pairwise comparisons matrices for all non-leaf nodes in a matrix panel; (ii) view the pairwise comparisons in a graph structure; (iii) view and edit the pairwise comparisons in a equalizer panel; (iv) locate and evaluate inconsistent judgments using Dissonance and Congruence matrices and Intransitive preference cycles in the matrix, equalizer and graph panels; (v) evaluate overall matrix consistency; (vi solve the problem using preference elicitation methods and view the final values of alternatives (vii) view the elicited weights for each criterion;) view alternatives final evaluation values

D. Ontology Management Module

The *ontology management* module is a Web application based on Ruby on Rails Framework that acts as a client application with respect to some Java Web Services in order to complete several tasks. The application uses Ruby specific tools and libraries for ontology files management, and calls to Web Services for interacting with the ontologies enclosed by these files.

a) Documenting ontologies: In order to offer an insight of the ontologies saved to the repository, the application comes with the feature of generating schemas and short descriptions of the components of each ontology. The system generates human-readable documentation having as an input OWL or RDF ontologies taking into consideration both ontological axioms and annotation. This comes as an improvement for accessing semantic information by a regular user using Web browsers, without being limited by Semantic Web specialist tools. *YOWL* is used for generating a documentation web-page from the ontologys files.

b) Querying the ontologies: The OWL API [8] is enacted to interacting with the ontologies loaded in the system. The interrogation is composed of pairs of requests and responses. When the list of individuals from a certain class is requested, the web service gets the values from two nodes, one is the ontology and the other one is the class from which the individuals must be listed.

c) Visual Representation of Ontologies: We enact the InfoVis tool for providing interactive web visualization of ontologies.

V. Applying AHP in the tourism domain

Our solution was tested for ontologies in the tourism domain. This section exemplifies how ontology related to tourism can be measured, searched, evaluated, and ranked using the Analytic Hierarchic Process.

A. Filtering available ontologies based on domain coverage

Assume the user specifies the following terms for domain coverage: *cruise, mountain, monument, museum, traveling, camping, hiking*. Given a repository of tourism ontologies, the domain coverage computed for each ontology against the above search terms is depicted in Fig. 3.

B. Enacting the AHP process

The solution contributions to *AHP* in the context of adaptation for ontology evaluation and selection are:

- extending the elicitation function to incorporate both positive, benefit criteria and negative, cost criteria in the same tree

TABLE II

ONTOLOGY EVALUATION METRICS.

Category	Name	Value	Description
Size	Classes	+	Number of concepts in the ontology
Size	Data Properties	+	Number of datatype properties between instances of classes and datatypes defined in the ontology. For example, *modelName* (String datatype) is the property of *Manufacturer* class.
Size	Object Properties	+	Number of relations between instances of two classes defined in the ontology. For example, *ownedBy* may be an object type property of the *Vehicle* class and may have a range which is the class *Person*.
Size	Individuals Count	+	Number of class instances.
Size	Logical Axioms	+	Number of assertions, including rules in a logical form that together comprise the overall theory that the ontology describes in its domain of application.
Consistency	Unsatisfiable Classes	-	Number of unsatisfiable classes. They cannot have instances for the ontology to remain consistent.
Consistency	Inconsistency	-	1, if ontology is inconsistent, 0 otherwise. An ontology is inconsistent when it violates its restrictions.
Cohesion	Avg SubClasses	+	The average number of subclasses for a class.
Cohesion	Avg SuperClasses	+	The average number of superclasses for a class.
Domain Coverage	Domain Coverage	+	Use-context dependent. The percentage in user given concepts (by words and their synonyms) found in an ontology.

Ontology Id	Ontology URI	Domain Coverage
102	http://rewerse.net/A1/otn/OTN.owl	0.2857
103	http://harmonisa.uni-klu.ac.at/ontology/skeleton.owl	0.0
104	http://www.info.uqam.ca/Members/valtchev_p/mbox/ETP-tourism.owl	0.1429
105	http://harmonisa.uni-klu.ac.at/ontology/moland.owl	0.1429
106	http://fivo.cyf-kr.edu.pl/ontologies/test/VOTours/TravelOntology.owl	0.1429
107	http://cui.unige.ch/isi/onto/2010/urba-en.owl	0.5714
108	http://en.openei.org/wiki/Special:ExportRDF/South_Africa_Department of Environment Affairs and Tourism	0.0
109	http://en.openei.org/wiki/Special:ExportRDF/Climate_Change_Adaptation and Mitigation in the Tourism Sector	0.0
111	http://jxml2owl.projects.semwebcentral.org/sample/tourism.owl	0.0
112	http://iri.columbia.edu/~benno/data_center.owl	0.0
113	http://www.pms.ifi.lmu.de/rewerse-wga1/otn/OTN.owl	0.2857
114	http://aabs-semanticweb-prototypes.googlecode.com/svn-history/r2/trunk/ontologies/2007/02/Test/needs.rdf	0.0
115	http://aabs-semanticweb-prototypes.googlecode.com/svn-history/r2/trunk/ontologies/2007/02/Flight/Flight.owl	0.0
116	http://aabs-semanticweb-prototypes.googlecode.com/svn-history/r2/trunk/ontologies/2007/02/Places/Places.owl	0.1429
117	http://www.esd.org.uk/standards/lgcl/1.03/lgcl-schema/lgcl.xml	0.0
118	http://www.cs.ox.ac.uk/isg/ontologies/lib/GardinerCorpus/http___protege.stanford.edu_plugins_owl_owl-library_travel.owl/2009-02-13/00120.owl	0.1429
119	http://harmonisa.uni-klu.ac.at/ontology/realraum.owl	0.0

Fig. 3. Domain Coverage for 17 ontologies in the tourism domain.

- adapting the Additive Normalization Method to calculate the Weighted Arithmetic Mean of alternative atomic measurements, allowing zero values in evaluation and avoiding unnecessary pairwise comparisons or fuzzy intervals

The AHP implementation used visual priority estimation tool *PriEsT*[3] to model the problem, interact with user for obtaining pairwise comparisons and a visual representation of preference inconsistencies, elicit weights using *Eigenvalue Method* and calculate elicitation accuracy. The *PriEsT* implementation has been modified to accommodate our proposed solution for alternative elicitation against atomic criteria (leaf nodes).

[3]http://sourceforge.net/projects/priority/

TABLE III

PREFERENCE ACCURACY FOR THE BEST ONTOLOGY.

Deviation	Value
Total direct deviation from direct judgments	2.33
Total indirect deviation from indirect judgments	29.23
Number of priority violations	0

TABLE IV

ELICITED WEIGHT FOR THE TOP LEVEL CRITERIA.

Criteria	Normalised Elicited Weight
Consistency	0.25
LanguageExpressivity	0.29
Size	0.22
Cohesion	0.12
DomainCoverage	0.12

C. Preference Consistency And Evaluation Accuracy

The results of our experiments in Tables III, IV, V and VI with ontologies in the tourism domain show the correlation between Pairwise Comparisons Matrix size, preference inconsistency and elicitation method accuracy. Table III shows the results of Eigenvector method accuracy of elicited criteria weights for user preference with small inconsistency, calculated using PriEsT [20] features . Tables IV, V and VI present criteria weights elicited from PC matrices using Eigenvector method.

Because large PC matrices are difficult to maintain consistent, small groups of sub-criteria are preferred, and the user is given suggestions to improve consistency using PriEsT features.

VI. RELATED WORK

We discuss related work from the perspectives of related (i) evaluation systems and (ii) ranking systems.

TABLE V
ELICITED WEIGHT FOR THE SIZE-RELATED SUB-CRITERIA.

Sub-Criteria	Normalised Elicited Weight
Classes	0.34
Object properties	0.18
Data properties	0.26
Individuals	0.15
Logical axioms	0.07

TABLE VI
ELICITED WEIGHT FOR THE CONSISTENCY-RELATED SUB-CRITERIA.

Sub-Criteria	Normalised Elicited Weight
Unsatisfiable concepts	0.83
Inconsistency	0.17

Ontology evaluation systems. OntoQA [22] uses a set of schema and structural metrics that evaluate how the classes, relationships and individuals are connected. The OntoQA system has been motivated by the SWETO ontology [22], intended to serve as test-bed for advance semantic applications. The metrics described by OntoQA refer to the general ontological properties and highlight key characteristics of ontology schema and their population. The metrics of OntoQA are divided in two main categories: 1) *Schema metrics:* evaluate ontology design and its potential for rich knowledge representation; respectively 2) *Instance metrics:* evaluate instances and data distribution within the ontology; measure how effectively the ontology is used to represent the knowledge modeled; OntoQA is useful for ontology users when considering an ontology as a source of information, but also for ontology developers that can evaluate their work when building an ontology.

The *OntoClean* tool [7] evaluates the ontologies at the metadata level, being domain independent. The formal properties of a taxonomy are assessed, providing means to derive measurable mismatches of a taxonomy with respect to an ideal structure by taking into consideration the semantic of "is-a" relationship. Hence, OntoClean is intended to improve the taxonomic structure of ontologies by providing relevant explanations of why mismatches occur.

OntoMetric [12] is an evaluation framework proposed to measure the suitability of existing ontologies. A taxonomy of 160 characteristics is proposed to allow users to choose and compare suitable ontologies. The metrics are divided into several domains, like tools, languages, methodologies, costs. The approach starts by defining an analytic hierarchy process that involves building a hierarchy tree having the objective of the problem as the root node. The intermediate nodes represent the criteria, while the lowest levels are the alternatives. Users can express preference between criteria. OntoMetric, besides having the capability of calculating some metrics, can also enrich the ontologies from different sources.

OntoRich framework [2] is a support tool for semi-automatic ontology enrichment and evaluation. The WordNet is used to extract candidates for dynamic ontology enrichment from RSS streams. With the integration of OpenNLP the system gains access to syntactic analysis of the RSS news. The enriched ontologies are evaluated against several qualitative metrics.

Ontology ranking systems. AktiveRank [1] is a technique for ranking ontologies based on the analysis of concept structures. It is integrated with SWOOGLE [4] to allow users to search for ontologies. The concepts that match user's request are identified by querying a set of terms. The measures involved in this ranking method are: class match measure, density measure, semantic similarity measure and betweenness measure. The ranking results are returned to the user as an OWL file, containing the ontology URIs and their total ranks. The total score is obtained by aggregating all the measures' values, by setting a weight to each of them.

A similar ranking approach is *OS_Rank* [27], a tool that ranks ontologies based on semantic relations and structure. As the previous method described, OS_Rank also uses SWOOGLE [4] for searching the ontologies that match some terms from the user queries. The ranking score is based on the following measures: class name, semantic relation and ontology structure. Like in the AktiveRank approach, this method can also be adjusted by the user by setting a weight to each calculated measure, according to its importance and relevance of criteria.

Another type of ranking method 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* [4] and *OntoKhoj* [17] 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. OntoKhoj search engine extends the traditional approach (keyword-based search) to cover the information in Semantic Web. This tool has multiple functionalities, including: advanced searching, ranking, aggregating and classifying of ontologies crawled from the Semantic Web. In the process of computing the final score, the following measures are involved: the concept name, content in special tags and literals pointed by a particular subject.

OntoRank algorithm is another ranking approach based on the link analyze method [6]. The authors of this system do not consider the user query an effective factor for ranking the results [21]. The importance of the ontology is evaluated in a static manner and two concepts are considered as a reference relationship only if there exists a relationship between instances of those classes. The final ranking is computed by adding the accessed probability of the ontology with the accessed probability of all imported ontology documents.

VII. CONCLUSIONS

Our AHP-based ontology evaluation system is an automatic decision-aiding software that evaluates ontologies based on their complex characteristics. The user decides the importance of each criterion in a simple manner, as well as the domain covered by the desired ontology.

The contributions of this paper are: Firstly, we proposed a hierarchical model of independent characteristics that describe ontologies. The hierarchy is used for analyzing the problem from different perspectives and at different abstraction levels. The decision is based on the concrete end-node measurements and their relative importance at more abstract levels. Secondly, we proposed a solution for evaluating the coverage of a user-specified domain based on semantic, as well as lexical similarity between terms, as domain coverage is an essential criterion considered in ontology reuse. Thirdly, we use the Analytic Hierarchy Process to diminish the complexity of the decision by allowing the user to pairwise compare the importance of only related attributes of the same abstraction level. The use of AHP brought benefits including group decision applicability, uniform treatment of qualitative and quantitative attributes [13], resemblance to human thinking by allowing inconsistency in expressing preference.

We argue that our solution is: i) automatic, ii) complex in suggesting a hierarchy of ontology attributes from abstract to concrete; iii) flexible in allowing user to express preferences between attributes, iv) allowing expressing relative preference in a natural, imprecise way and computing precise global preference, v) precise in terms of evaluation, vi) informative, allowing the user to see the arguments and explanations behind the decision.

Our current work has two related directions: Firstly, we aim to apply our AHP-based ranking method at the Ontology Building Competition [9]. Secondly, in line with [11] we try extend AHP-decision with argumentation capabilities in a multi-agent setting.

Acknowledgments

We thank the reviewers for their valuable comments. This work was supported by the Romania-Moldova Bilateral Agreement entitled "ASDEC: Structural Argumentation for Decision Support with Normative Constraints", Adrian Groza is also 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, 2006, pp. 5–9.

[2] 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.

[3] V. Belton and T. Gear, "On the meaning of relative importance," *Journal of Multi-Criteria Decision Analysis*, vol. 6, no. 6, pp. 335–338, 1997.

[4] 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.

[5] I. Durbach, R. Lahdelma, and P. Salminen, "The analytic hierarchy process with stochastic judgements," *European Journal of Operational Research*, vol. 238, no. 2, pp. 552 – 559, 2014.

[6] T. Finin, L. Ding, R. Pan, A. Joshi, P. Kolari, A. Java, and Y. Peng, "Swoogle: Searching for knowledge on the semantic web," in *Proceedings of the National Conference on Artificial Intelligence*, vol. 20, no. 4. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2005, p. 1682.

[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] I. Jimborean and A. Groza, "Ranking ontologies in the ontology building competition boc 2014," in *IEEE 10th International Conference on Intelligent Computer Communication and Processing (ICCP2014), Cluj-Napoca, Romania, 4-6 September 2014*. IEEE, 2014, pp. 75–82.

[10] R. C. B. Leonelli, "Enhancing a decision support tool with sensitivity analysis," Master's thesis, University of Manchester, Faculty of Engineering and Physical Sciences, 2012.

[11] I. Letia and A. Groza, "Arguing with justifications between collaborating agents," in *Argumentation in Multi-Agent Systems*, ser. Lecture Notes in Computer Science, P. McBurney, S. Parsons, and I. Rahwan, Eds. Springer Berlin Heidelberg, 2012, vol. 7543, pp. 102–116. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-33152-7_7

[12] 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.

[13] A. Marginean, I. A. Letia, and S. Zaporojan, "Using domain specific hierachical good practice for ranking service composition," in *16th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, Timisoara, Romania, 22-25 September, 2014*.

[14] 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.

[15] W. Ouerdane, N. Maudet, and A. Tsoukis, "Argumentation theory and decision aiding," in *Trends in Multiple Criteria Decision Analysis*, ser. International Series in Operations Research & Management Science, M. Ehrgott, J. R. Figueira, and S. Greco, Eds. Springer US, 2010, vol. 142, pp. 177–208.

[16] 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.

[17] 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.

[18] T. L. Saaty, "Decision making with the analytic hierarchy process," *International journal of services sciences*, vol. 1, no. 1, pp. 83–98, 2008.

[19] T. L. Saaty and L. G. Vargas, "Comparison of eigenvalue, logarithmic least squares and least squares methods in estimating ratios," *Mathematical Modelling*, vol. 5, no. 5, pp. 309 – 324, 1984.

[20] S. Siraj, R. C. Leonelli, J. A. Keane, and L. Mikhailov, "Priest: a tool to estimate priorities from inconsistent judgments," in *Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on*. IEEE, 2013, pp. 44–49.

[21] R. Subhashini, J. Akilandeswari, and V. Sinthuja, "A review on ontology ranking algorithms." *International Journal of Computer Applications*, vol. 33, 2011.

[22] 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.

[23] A. Tsoukiás, "From decision theory to decision aiding methodology," *European Journal of Operational Research*, vol. 187, no. 1, pp. 138 – 161, 2008.

[24] O. S. Vaidya and S. Kumar, "Analytic hierarchy process: An overview of applications," *European Journal of operational research*, vol. 169, no. 1, pp. 1–29, 2006.

[25] W. Wei, "Enhancing PriEsT with group decision making," Master's thesis, University of Manchester, Faculty of Engineering and Physical Sciences, 2013.

[26] J. Yu, J. A. Thom, and A. Tam, "Requirements-oriented methodology for evaluating ontologies," *Information Systems*, vol. 34, no. 8, pp. 766 – 791, 2009, sixteenth {ACM} Conference on Information Knowledge and Management (CIKM 2007).

[27] 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.