Topic-Specific Trust and Open Rating Systems: An Approach for Ontology Evaluation

Holger Lewen
Institute AIFB
University of Karlsruhe
76131 Karlsruhe, Germany
lewen@aifb.uni-karlsruhe.de

Kaustubh Supekar, Natalya F. Noy, and Mark A. Musen
Stanford Medical Informatics
Stanford University School of Medicine
Stanford, CA 94305-5479, USA
{ksupekar, noy, musen}@stanford.edu

ABSTRACT

To achieve better interoperability among intelligent applications, and to relieve knowledge engineers from the burden of developing ontologies from scratch, it is critical to reuse ontologies. However, there are two main reasons why the reuse of ontologies is rare: (1) current ontology repositories allow only simple keyword-based search facilities, and (2) even when a user finds an ontology, the information about the ontology quality and (re)usability is not available.

In this paper, we present an Open Rating System based approach for ontology evaluation. The core Open Rating System model is extended with topic-specific trust to provide more accurate personalized ontology rankings. Our model is partially implemented in Knowledge Zone—a web-based ontology repository where users can submit their ontologies, annotate them with metadata, search for existing ontologies, find out their rankings based on user reviews, post their own reviews, and rate reviews.

1. INTRODUCTION

In recent years, there has been a widespread use and application of ontologies for data annotation, for data integration, and for building intelligent applications [9]. However, most of these ontologies are developed from scratch by individuals and institutions. As a result, there is an assortment of ontologies of varying quality and this ontologies do not interoperate easily. Moreover, ontology development is a non-trivial task that requires substantial investment of time and resources. Reuse of ontologies will not only achieve better interoperability among applications but also will relieve knowledge engineers from developing ontologies from scratch.

Current approaches (such as Swoogle,[1] OntoSearch,[2] the OBO ontology repository,[3] or the Protégé OWL ontology library[4]) aiming to facilitate ontology reuse have focused on developing ontology repositories that are mere listings of ontologies. These approaches provide keyword-based search facilities, enabling users to search through possibly thousands of ontologies in the repository and returning ontologies that have that keyword. However, a simple keyword-based approach suffers from both poor precision and poor recall. For example, a user query for the keyword "anatomy" in one of the popular ontology search engines (Swoogle [2]) produces 59 "hits." It is then left to the user to scour through these 59 XML files and to evaluate them to find the ontology that is best suited for his purpose. The task of evaluating ontologies is difficult as these ontologies do not embed metadata information such as the intended purpose of the ontology, the maturity of its content, the level of support, semantic (logical consistency) and syntactic correctness, and so on. This subjective information about the ontology is critical while evaluating an ontology. Furthermore, lack of metadata also substantially effects the recall of search results for keyword-based search facilities. A search for the keyword "anatomy" does, for example, not return the GALEN ontology, which is one of the more popular anatomy ontologies, even though it is crawled by the Swoogle search engine. If keywords describing the subject matter of an ontology were stored as part of the ontology metadata, GALEN would have been returned as the result of the query for "anatomy."

We have developed Knowledge Zone—a web-based ontology repository where users can submit their ontologies and annotate ontologies with metadata information. These metadata elements are features that characterize an ontology and have been organized in a metadata ontology.[5] This metadata information is used to drive a structured query interface, where users can build queries allowing them for example to find anatomy ontologies that have been used for data integration, that are developed in OWL, that are available under the GNU license.

As more ontologies become available, even structured and specific queries will yield several relevant ontologies in their search results. To enable users to evaluate these ontologies, we provide ontology *rankings* along with the other metadata information associated with the ontology. Rankings are based on peer-reviews of an ontology, which are entered by other users, and are also based on ratings of these reviews. Users not only can provide reviews on the ontology as a whole, but also can provide reviews on different dimensions of the ontology, such as degree of formality, maturity, quality of content or reusability. Open Rating Systems provide means for product evaluation by having potentially any user write reviews on products and other users judge the helpfulness of these reviews. We have developed a novel model that

[1]http://swoogle.umbc.edu/
[2]http://www.ontosearch.org/
[3]http://obo.sourceforge.net/
[4]http://protege.stanford.edu/plugins/owl/ontologies

WWW2006, May 22–26, 2006, Edinburgh, UK.

[5]http://tinyurl.com/qfp2s

extends the traditional Open Rating System (ORS) model [5] with topic-specific trust, to compute ontology rankings based on (1) reviews on the ontology as a whole, (2) reviews on different dimensions of the ontology, and (3) ratings of these reviews.

This paper is organized as follows. Section 2 describes related work, while section 3 covers a traditional Open Rating System model with a Web of Trust and its limitations. Section 4 describes our extended ORS model with topic specific trust, and its applicability to ontology evaluation. Section 5 deals with incorporating our extended model in Knowledge Zone, and finally, we present our plan to evaluate our model for ontology evaluation.

2. RELATED WORK

Two recent approaches for ontology evaluation [2, 12] have proposed a page-rank based algorithm to rank ontologies. In a page-rank based approach ontologies are treated as webpages and the rank of an ontology is computed as a function of number of outgoing links and the number of incoming links. As a result, ontologies that are reused more often get a higher ranking than others. However, these approaches do not work effectively today because the reuse of ontologies is rare; consequently, for a given ontology there are only a small number of incoming and outgoing links.

Other works [15] treat the ontology as a black box and compute the rank of an ontology as a function of ontology metrics such as number of classes, number of properties, depth and width of an ontology, and so on. On similar lines there are methods that evaluate the quality of conceptual models [17, 4, 16] and methods that measure the usability in an application context [8]. Brank and Grobelnik provide a survey of existing ontology evaluation techniques[1].

We argue that even though it is important to know the quality of an ontology as a function of quantifiable metrics, it is equally important to know subjective information associated with an ontology. Hartmann et al [7] published a set of metadata similar to ours for improving ontology search. However, their repository Onthology[6] and P2P-client Oyster[7] do not allow ontology evaluation yet.

Our approach is similar to the one described by Noy and colleagues [10] wherein they have proposed to use the traditional ORS model and a Web of Trust to compute ontology rankings based on subjective metadata provided by the user. While their paper presented a convincing motivation for the need of subjective ontology evaluation and therefore the use of ORS, it does not support topic-specific trust and detailed ratings.

3. OPEN RATING SYSTEMS

The basic idea of Open Rating Systems[5] is to have a democratic approach to rating where anyone can review pieces of content. The real power of this approach lies in the concept of **metarating**: users can rate not only the content itself but also reviews of the content provided by others. The reader is probably familiar with "Was this review helpful to you" button on many rating sites. This concept has proven highly successful and is currently employed at Epinions,[8] Slashdot,[9] Amazon[10] (in the user review section), iTunes[11] (to review music), and other sites.

Based on the feedback a user provides to the Open Rating System (usually by commenting on the helpfulness of existing reviews), a ranking of products and reviews of those products is presented. This ranking is user-specific and therefore allows a personalization of the way content in the system is presented to the user.

3.1 Traditional Open Rating System

While all the technical details of the traditional ORS model can be found elsewhere [5, 6], we will present an adapted view on them. We will explain what an ORS model without extensions would look like when adapted to ontology evaluation (as suggested in [10]) and will present its shortcomings in that context. We will then compare it to our extended ORS model in section 4 to justify the need for topic-specific trust.

3.1.1 Model

A basic model of an ORS for ontologies consists of 6 components:

1. The ontologies $O : \{O_1, O_2, O_3, \cdots, O_{N_1}\}$ that are evaluated (reviewed).
2. The agents $A : \{A_1, A_2, A_3, \cdots, A_{N_2}\}$ that participating in ORS. The agents are users either evaluating an ontology or rating the trustworthiness of other users (implicitly by commenting on the helpfulness of reviews).
3. A value set $D : \{D_1, D_2, D_3, \cdots, D_{N_3}\}$ of possible ratings of ontologies (e.g. 1 star, 2 stars etc.).
4. A value set $T : \{T_1, T_2, T_3, \cdots, T_{N_4}\}$ of possible ratings of agents by other agents. Most of the current ORS follow the approach of counting a statement on the helpfulness of a review of an author as a trust rating. In most existing Open Rating Systems, T is limited to $T = \{positive, negative\}$ (helpful, not helpful).
5. A partial function $R : A \times O \to D$. This function stores the ratings agents give to ontologies. R will normally be very sparse, because most agents will rate only a small number of ontologies, if any at all.
6. A partial function $W : A \times A \to T$. It stores the ratings of agents on other agents and will normally also be very sparse.

3.1.2 Trust

While trust can have many definitions depending on the context it is used in, we define trust in Open Rating Systems as follows: "Trust in Open Rating Systems corresponds to the feeling that the information delivered by a certain reviewer will be correct and useful." In Open Rating Systems, a trust statement is normally made by clicking whether a given review was helpful or not. An advantage of having a platform where users provide information about their trust connection to other users is the possibility to use that information to infer additional trust connections by means of

[6] http://www.onthology.org/
[7] http://oyster.ontoware.org/
[8] www.epinions.com
[9] www.slashdot.org
[10] www.amazon.com
[11] www.apple.com/itunes

trust propagation. Taking users as nodes and trust statements as edges, a trust graph can be inferred, often referred to as Web of Trust. In order to perform propagation methods on that graph, certain properties of trust have to be assumed. Jennifer Golbeck [3] emphasized four properties as being central:

1. **Transitivity**: The basic idea of transitivity in trust models ([5][6][3]) is that if A trusts B and B trusts C, A should also trust C to some extent. The question on how trust should decay when being propagated is a modeling parameter and can vary depending on application needs.

2. **Composability**: The general idea of composability of trust is that if people get the same recommendation from different trusted sources, the trust assigned to that recommendation will be higher than if only one source is available.

3. **Personalization**: When dealing with humans interacting in trust networks, instead of computer agents, everybody will have his own idea of what trust is and whom to trust. The very same person will be trusted by some and mistrusted by others. Personalization is a very important aspect of ORS. Most of the times, reviews will be subjective and based on the authors beliefs and preferences. One way of achieving at least a sensible ranking of reviews is taking into account the Web of Trust generated based on feedback on reviews given by each user.

4. **Asymmetry**: The concept of asymmetry of trust is of significant importance when modeling trust in very anonymous settings like the WWW. While A might find B's reviews helpful, B might not even know A. Therefore it makes no sense to assume all trust relationships would have to be mutual.

3.1.3 Propagation of Trust and Distrust

The real strength of Web of Trust based approaches comes into play when propagation algorithms are run on the graph. Guha and colleagues [6] developed a framework to propagate trust in a Web of Trust with the goal to "predict an unknown trust/distrust value between any two users" based on existing trust and distrust information.

In the framework, for a universe of n users, there are two global matrices storing trust and distrust. The trust matrix T has entries $t_{ij} \in [0, 1]$ meaning that user i trusts j with value t_{ij}. Analogously, the distrust matrix D has entries $d_{ij} \in [0, 1]$ expressing the distrust between user i and user j.

The propagation of trust and distrust starts with a generic *belief matrix* B, which is based on T and D. The specific way in which B is composed can differ depending on the implementation and desired propagation behavior of mistrust. There are four different basic propagation techniques, referred to as atomic propagation steps (see Table 1).

The propagation of distrust poses an interesting problem because there is no single right way to model it. Distrust is not necessarily transitive. If A does not trust B and B does not trust C, it is not clear whether A should trust C or not. Guha and colleagues [6] present and evaluate different possibilities of modeling distrust. According to their evaluation, using single step distrust propagation (if A distrust B, B distrusts C and C distrusts D, it is assumed that A will distrust C, but no assumption is made about the relationship between A and D) produces best results. For the rest of the paper, we will set $B = T$ according to the requirements of single step distrust propagation, which is called single step distrust propagation since the distrust is not propagated along with trust, but only 1 step each iteration (see equation 2).

Table 1: Atomic Trust Propagation

Propagation	Operator	Description
Direct Propagation	B	If A trusts B, someone trusted by B should also be trusted by A
Co-Citation	$B^\top \cdot B$	If A trusts B and C, someone trusting C should also trust B
Transpose Trust	$B^\top$	If A trusts B, someone trusting B should also trust A
Trust Coupling	$B \cdot B^\top$	If A and B trust C, someone trusting A should also trust B

In order to infer trust relationships in the normally poorly connected Web of Trust, a combination of all atomic trust propagation techniques forming the combined matrix $C_{B,\alpha}$ is used:

$$C_{B,\alpha} = \alpha_1 B + \alpha_2 B^\top B + \alpha_3 B^\top + \alpha_4 B B^\top \tag{1}$$

where $\alpha = (\alpha_1, \alpha_2, \alpha_3, \alpha_4)$ is a vector representing weights for combining the four atomic propagation schemes, B is the belief matrix and $B^\top$ is the transposed belief matrix. Entries in $C_{B,\alpha}$ indicate how trust can be propagated within the Web of Trust.

To propagate the trust, it is necessary to apply $C_{B,\alpha}$ on the initial trust information available. Let $P^{(k)}$ be a propagation matrix where each entry describes how strong the trust is between users after k propagation steps:

$$P^{(k)} = C_{B,\alpha} \cdot (T - D) \tag{2}$$

Using a combination of $P^{(k)}$ with different propagation depths, a final propagation matrix F can be computed using a weighted linear combination:

$$F = \sum_{k=1}^{K} \gamma^k \cdot P^{(k)} \tag{3}$$

where K is a suitably chosen integer and γ is a constant that is smaller than the largest eigenvalue of $C_{B,\alpha}$. K represents the maximal depth of trust propagation in the Web of Trust, γ is a parameter basically determining the rate of decay of trust as propagated within the Web of Trust (the further trust is propagated, the weaker it becomes).

While it is mathematically sound to perform the computation of trust values on a continuous scale, at some point, those values have to be interpreted as trust or distrust. The most successful method presented by Guha and colleagues [6] is called "majority rounding". The basic idea is to use information from the original belief matrix B to make assumptions about whether an inferred trust value should be interpreted as trust or distrust. Suppose a user i expresses trust and distrust for n people (entries in the trust matrix T or distrust matrix D), and we need to infer a trust relationship towards a user j. Using the final propagation matrix F, all inferred trust values linking i to the n users initially trusted or distrusted are sorted in the ascending order, including the entry f_{ij}. Then, depending on the local neighborhood of trust statements in the ordered set, f_{ij} is interpreted as trust or distrust, based on the majority of trust statements in the neighborhood.

Based on the inferred trust information, a ranking of reviews of other users can be made. We will go into details on how we perform such a ranking in section 4.

3.2 Limitations of Traditional ORS-Model

One of the main limitations of the traditional model of an ORS is the fact that trust or distrust can be assigned only *globally*. There is no way to trust a reviewer in a domain X (e.g. Science) but to distrust him in a domain Y (e.g. Health). Suppose an agent A_1 evaluates ontologies some of which cover domain X and some cover domain Y. Suppose another agent A_2 trusts the reviews A_1 provides for ontologies in domain X, but distrusts his reviews for ontologies in domain Y. These two trust/distrust values would cancel each other out to some extent instead of being treated as two distinct statements because the model captures trust between *agents* and not trust specific to statements made by agents ($W : A \times A \rightarrow T$). So the system would be able to capture information that Alice trusts Bob, but not that Alice trusts Bob in the Science domain. In the context of ontology evaluation, the current system would not allow a user to specify that reviews of a certain reviewer should only be trusted when covering ontological aspects that belong to a domain of expertise of that reviewer (e.g. only trusting the reviewer about reviews on the usability of ontologies covering medical content).

Another limitation of the traditional model is the lack of a link between a review and the trust assigned based on that particular review. The system does not store the information that a user A liked a particular review B of user C. As a result, a bad review by a normally good reviewer would still be ranked higher than deserved.

Maybe one of the biggest limitation is the fact that ontologies can only be rated as a whole ($R : A \times O \rightarrow D$), and overall ratings are not dynamically created using existing reviews based on a user's preferences. So the current Open Rating System model would not allow to just review parts of an ontology or certain aspects of it.

4. INTRODUCING TOPIC-SPECIFIC TRUST IN OPEN RATING SYSTEMS

To address all problems mentioned above, we developed an extended Open Rating System model including topic-specific trust, which we will refer to as Topic-Specific Open Rating System (TS-ORS) for the rest of the paper.

4.1 TS-ORS model for Ontology Evaluation

Our TS-ORS model consists of 11 components:

1. A set of ontologies: $O : \{O_1, O_2, O_3, \cdots, O_{N_1}\}$
2. A set of domain concepts: $C : \{C_1, C_2, C_3, \cdots, C_{N_2}\}$ (not to be confused with concepts of the actual ontologies O). Those domain concepts could be DMOZ[12] concepts like "Science" or "Health". They denote the domain an ontology tries to capture or can be categorized in.
3. A taxonomy (like DMOZ) of domain concepts C called H_C: concepts are related by the directed, acyclic, irreflexive transitive relation H_C, ($H_C \subset C \times C$). $H_C(C_i, C_j)$ means C_i is a subconcept of C_j
4. A "1 to n"-relation L (type of) linking O to C: $L : \{O_i \rightarrow \{C^*\}\} \mid O_i \in O, C^* \subseteq C$ This is linking for example ontology O_i to domains "Science" and "Health".
5. A set $X : \{X_1, X_2, X_3, \cdots, X_{N_3}\}$ of ontology properties (like name, domain coverage, usability, maturity)
6. A set $A : \{A_1, A_2, A_3, \cdots, A_{N_4}\}$ of agents participating
7. A set $D : \{D_1, D_2, D_3, \cdots, D_{N_5}\}$ of possible ratings of ontologies, divided into two subsets D_m for ratings intended for the machine to interpret (star ratings) and D_n for ratings not intended for the machine to interpret (free text reviews).
8. A relation P assigning possible ratings D to X: $P : \{X_j \rightarrow \{D^*\}\} \mid X_j \in X, D^* \subseteq D$ This is important because some properties may require different rating scales or may not be rated sensibly at all (e.g. name, author).
9. A set $T : \{T_1, T_2, T_3, \cdots, T_{N_6}\}$ of possible trust ratings
10. A partial function $R : A_i \times O_j \times X_k \rightarrow D^{O_j} \mid [A_i \in A, D^{O_j}$ consisting of D_m and explanation D_n, $(X_k \rightarrow D^{O_j}) \in P]$. This function stores the ratings of an agent on a certain property of an ontology.
11. A partial function $W : A_n \times R \rightarrow T_m \mid A_n \in A, T_m \in T$. It stores the ratings of agents on specific evaluations of other agents (e.g. helpful / not helpful)

R now allows a user to review certain properties of an ontology rather than reviewing the whole ontology in a global review. An evaluation of a property of an ontology can be seen as a combination of a machine interpretable rating and a justification of that rating for the human user. Since W now captures the information which evaluation a trust statement was made on, trust can be assigned more specifically. In the following section, we will introduce the intended ways to express trust.

4.2 Trust Statements in TS-ORS model

The intended way trust statements should be made in this model is expressing trust by commenting on the helpfulness of an evaluation of an ontology property (see definition for W). However, it would take a user a long time to review all evaluations written by a given reviewer. Users will want some means of assigning trust independently from evaluations, on a more general level. To address this problem, we allow to express trust on coarser levels of granularity (see Table 2) as a shortcut. It is important to note that those statements are only meant to be shortcuts to making a lot of

[12]www.dmoz.org

single statements W on specific evaluations. However, since our computations are performed at the lowest captured level of granularity, those shortcut statements have to be resolved to single W statements. This can be done by automatically replacing statements covering a greater scope of ontologies and ontology properties with a number of W statements covering all ontologies and properties within the scope of that statement (using the same T_u value). In case of contradicting statements, which can occur if a user states to trust another user to evaluate ontologies in general but not a certain property, more precise statements are not overwritten. That means an existing W statement will not be overwritten with the value of W statements produced when resolving a shortcut trust statement (more precise statements are more important).

4.3 Computing Trust Values for the Ranking

Now that all trust statements are in the form W, trust ranks can be computed. Note that in contrast to the traditional model, we do compute trust relationships for every property of every ontology (every $O_n X_k$ combination) specifically. We have to distinguish two possibilities when a ranking of evaluations of a property of an ontology has to be computed for a user querying the system: Either that user has made a specific trust statements for any available evaluation covering that $O_n X_k$ combination or not. If a user has not made any trust statement, no local trust information can be inferred. In that case, the ranking has to be based on all trust statements (made by all other users) affecting that $O_n X_k$ combination. This is done by using a modified version of the TrustRank (4) and DistrustRank (5) algorithms introduced in [5]:

$$TR_{N+1}(A_u) = (1-d) + d \cdot (\sum\nolimits_{v \in T_v} \frac{TR_N(v)}{N_v}) \qquad (4)$$

where TR is short for $TrustRank$, A_u is the agent whose TrustRank is computed, $v \in T_v$ is the agent trusting A_u, N_v is the total number of agents agent v trusts, d is a damping factor between 0 and 1 (usually set to 0.85), and N is the number of iterations.

$$DistrustRank(A_u) = \sum\nolimits_{v \in B_v} \frac{TrustRank(v)}{N_v} \qquad (5)$$

where A_u is the agent whose DistrustRank is computed, $v \in B_v$ is the agents distrusting A_u, N_v is the total number of agents the agent v distrusts. Intuitively speaking, TrustRank assigns trust to agents based on how many other agents trust them and how important the opinion of those agents is. The same holds true for DistrustRank, it is taking into account who distrusts an agent and how important the distrusting agents are.

As it is evident, TrustRank is basically just a PageRank [11]. In contrast to TrustRank, DistrustRank can be computed with only one iteration of the algorithm.

If local trust information is available, propagation of trust along a user's web of trust can be performed. Guha and colleagues performed an extensive evaluation, testing their algorithm using real world data and provided valuable insights towards the best choice of parameters [6]. Because the algorithm was proven to produce good results on real-world data, we use the same parameters in our computation of the final propagation matrices F. We compute a propagation matrix F for every $O_n X_k$ combination featuring evaluations. We perform the calculation of $F_{O_n X_k}$ using single-step distrust propagation and majority rounding (see section 3.1.3) as follows:

1. $B_{O_n X_k} = T_{O_n X_k}$
2. $C_{B_{O_n X_k}, \alpha} = 0.4 \cdot B_{O_n X_k} + 0.4 \cdot B^{\top}_{O_n X_k} B_{O_n X_k} + 0.1 \cdot B^{\top}_{O_n X_k} + 0.1 \cdot B_{O_n X_k} B^{\top}_{O_n X_k}$
3. $P^{(k')}_{O_n X_k} = C^{k'}_{B_{O_n X_k}, \alpha} \cdot (T_{O_n X_k} - D_{O_n X_k})$
4. $F_{O_n X_k} = \sum_{k'=1}^{7} 0.9^{k'} \cdot P^{(k')}_{O_n X_k}$
5. Interpret values using "Majority Rounding"

where $T_{O_n X_k}$ and $D_{O_n X_k}$ are Trust and Distrust matrices (as defined in section 3.1.3) specific to the $O_n X_k$ combination, and K is set to 7 (since it is not sensible to propagate trust further than 7 steps). Explanations of the different matrices and operations can be found in section 3.1.3

4.4 Ranking Evaluations at the Property Level of an Ontology

Evaluations that exist for an $O_n X_k$ combination are linked to their author. The quality of an evaluation is determined by feedback from the user community on how helpful it was. The TrustRank, DistrustRank and F values provide the information about the global ranking of authors (TrustRank and DistrustRank) and about the ranking of authors as perceived by each single user (F).

The first choice a user has to make when querying the system is how to combine TrustRank and DistrustRank. Some may tend to put a great emphasis on TrustRank values while others rely on the significance of DistrustRanks. Each user therefore has to choose a parameter $\alpha \in [0, 1]$ that can be stored in a profile and is used to compute:

$$CombinedRank(A_i) = TR(A_i) - (\alpha \cdot DR(A_i)) \qquad (6)$$

where TR is short for $TrustRank$ and DR is short for $DistrustRank$. When evaluations have to be ranked, the system first looks if any local trust information exists for the user - $O_n X_k$ combination. If yes, using $F_{O_n X_k}$, the information who is trusted and who is distrusted is retrieved and then both trusted and distrusted users are ordered using their inferred local trust rank. In case two users share the same local trust value, the order of those reviewers is determined by their $CombinedRank$. The results of the ranking will start with the users that are trusted locally. When the ranks of all the locally trusted users have been determined, the following ranks are filled with all reviewers that no local trust information is available for, using their $CombinedRank$. Lastly, the locally distrusted reviewers are ranked starting with the least distrusted reviewer ending with the most distrusted reviewer. If no local trust information is available, ranking is solely based on $CombinedRank$. At the end of the ranking of the reviewers, their reviews are presented to the user in that order. Note that local ranks always override global ranks, so that a user having a very subversive view will have reviews by his favorite reviewers ranked first instead of reviews that the majority of users like.

Table 2: Allowed Trust Statements and Their Scope

Statement	Scope	Explanation
W	$A_i \times A_j \times O_n \times X_k \times D^{O_n} \to T_u$	Statement on a specific property of a specific ontology
$W_{O_n X}$	$A_i \times A_j \times O_n \to T_u$	Statement on all properties of a specific ontology
$W_{C_n X_k}$	$A_i \times A_j \times C_n \times X_k \to T_u$	Statement on a specific property of all ontologies in a specific category
$W_{C_n X}$	$A_i \times A_j \times C_n \to T_u$	Statement on all properties of ontologies in a specific category
$W_{C X_k}$	$A_i \times A_j \times X_k \to T_u$	Statement on a specific property of all ontologies
W_{CX}	$A_i \times A_j \to T_u$	Statement on all properties of all ontologies

4.5 Computing an Overall Evaluation of an Ontology

Each ontology in our repository has several properties it can be evaluated on, such as degree of formality, maturity, quality of content or reusability. Combining the ratings provided in the context of evaluation of its properties, an overall rating can be inferred for an ontology. It is important to note that there is no single right way to combine the evaluations of an ontology's properties. Depending on the intended application, different aspects may be important to the user. Therefore, for every query, weights μ_k (that are normalized to ensure that $\sum \mu_k = 1$) have to be assigned to all ontology properties X_k that the system should take into account for computing the overall rating. A user who wants to find ontologies that have a high maturity and reusability might choose to assign $\mu_1 = 0.5$ to property "maturity" and $\mu_2 = 0.5$ to property "reusability". While we assume most users searching our repository will know exactly which ontology properties are important for their intended use of the ontology, μ_k-weight-presets will be offered for the rest. Since every ontology property X_k will have one top-ranked evaluation (specific to the user querying) featuring a rating D_m associated by R, an overall rating for an ontology can be computed as $D_{O_n} = \sum \mu_k \cdot D_m^{O_n X_k}$. Since the system uses parameters specific to each user, D_{O_n} can not be preprocessed. In contrast to the traditional model, it is now possible to compose an overall rating using evaluations of different reviewers.

4.6 Ranking Ontologies

The main tasks the Open Rating System has to perform in our ontology repository is ranking ontologies that show up as result of a query or when browsing categories in the domain hierarchy. The first step in getting a ranking is finding the objects that should be ranked. In the case of a query, a simple pattern-based comparison of the search term and metadata annotation in the system should provide a subset of ontologies that have to be ranked. If the task is to rank all ontologies belonging to a certain domain concept, the concept hierarchy H_C is traversed down adding all ontologies that are defined to be instances by L at each concept C_i to the result space. An example would be a user browsing the science domain. First all ontologies being science ontologies would be added to the result space, than the domain hierarchy would be traversed down adding all biological, chemical, computer science a.s.o. ontologies until all ontologies covering science or any subcategory are added. Once all ontologies that have to be ranked have been found, they are ordered using the inferred overall rating D_{O_n} (see section 4.5). The ranking results are highly user-specific, because they are based on a user's trust statements, the parameter α, and the weights μ_k assigned to the different ontology properties.

4.7 Value Added in Terms of Ontological Evaluation

The traditional Open Rating System model allowed only evaluations of ontologies as a whole. This is insufficient for a construct as complicated as an ontology. People searching for ontologies to reuse in their application need very detailed feedback on multiple ontology properties. This can only be provided if ratings are permitted on single ontology properties. Furthermore, the knowledge and experience needed to give a profound rating on a certain aspect of an ontology is considerable. In many cases, reviewers are only qualified to evaluate certain properties of an ontology. It would not to be sensible to ask those reviewers to evaluate a complete ontology. Reviewers will very likely only have certain areas of expertise. They could for example be experts in a specific domain and therefore be qualified to rate the domain coverage of an ontology that tries to model this domain. Others might have used an ontology and thus qualify for providing a rating on its reusability. Personalization is also very important for ontology retrieval, because a user's needs and preferences will be very specific and individual.

Our proposed TS-ORS model solves addressed problems. Users or ontology experts can evaluate exactly that property of an ontology they have expert knowledge on. Users searching for an ontology to reuse can combine those rankings using individual weights for different ontology properties. If users overestimate their knowledge and write reviews of poor quality, those will be ranked low automatically by the system if enough other users state they were not helpful. In contrast to the traditional model, it is now possible that a reviewer will have a high distrust rank in one domain and a high trust rank in another. Distrust and trust statements in different domains do not affect each other. Since there is no global trust value anymore, credit is only given where it is deserved and weak reviews for some ontologies will not discredit good reviews for others. One particular problem of Open Rating Systems targeting a very small community is that many users will know each other personally or by reputation. Those personal bonds might cause users to be reluctant to state their true opinion if they have to criticize a colleague or even friend. With the new model it is possible to just point out certain weak points of an ontology without discrediting it as a whole. It is easier to just state that a concept has been embedded at the wrong place in the concept hierarchy than to say the whole ontology is bad because of errors in the concept hierarchy. Given a user has entered enough trust statement and sensible weights, the system will provide a highly personalized view and very specific ranking results. The increased computational complexity of the new model is negligible in the specific ontology repository scenario because the number of ontologies, ontology properties and agents will most likely be relatively small compared to big commercial applications used by Amazon or Apple. That means that once the new model is fed with trust statements and enough reviews and ontologies are entered into Knowledge Zone, the ranking quality will significantly increase in comparison to the traditional model. Users will benefit from the possibility to specify their personal preferences when searching and domain experts will be elected by the users "on the fly" by rating the helpfulness of their reviews.

5. KNOWLEDGE ZONE - OUR APPROACH TO ONTOLOGY REPOSITORIES

We have developed Knowledge Zone—an environment for submitting ontologies to a repository, annotating ontologies with metadata, and providing reviews of ontologies along different dimensions. We plan to use Knowledge Zone as a testbed for the evaluation of the topic-specific trust model (Figure 1).

The content and behavior of the system is guided at runtime by the Metadata Ontology[13]. This ontology comprises all information about what metadata can be entered, how the review possibilities look like, and how help can be provided. We invite the interested reader to take a look at the OWL ontology itself. More technical details can be found in [13].

By taking a look at the Metadata Ontology, the ontology properties (X in TS-ORS model) can be seen. We chose those properties because they capture information that is easy to provide while submitting an ontology and at the same time capture most of the formal questions user will ask about an ontology when considering reuse. Additional information on the Metadata Ontology can be found in [14].

Rating and Ranking: For the evaluation phase, we allow two kinds of reviews. On the one hand, a "short" review which basically allows to enter a complete evaluation of an ontology compressed into a single rating. On the other hand, a "detailed" review which allows an elaborate evaluation of an ontology, consisting of evaluations on different ontology properties. Once the evaluation phase is completed, we will restructure the detailed review to be consistent with the new model (allowing star ratings in combination with an explanation for the human user for relevant ontology properties).

Interoperability with Other Ontology Repositories: We plan to expose all of Knowledge Zone's functionality via a web service interface, therefore relieving authors of the burden of submitting their ontology to different repositories. This way, other ontology repositories like Swoogle, Onthology or its P2P-version Oyster can include all ontologies submitted to Knowledge Zone by just gathering its data using the web service.

Integration of TS-ORS Model into Knowledge Zone: Mapping the formal model defined in chapter 3.1.1 to our application is rather straight-forward. The ontologies (O) are added into the repository by either their authors or by a user, who, in the process of submitting it, provides metadata information about the domain (C) the ontology covers (L). We use the DMOZ[14] concepts to describe the ontology domain in order to build on the popular[15] DMOZ taxonomy (H_C) instead of building our own taxonomy. The ontology properties (X) are provided by the metadata ontology and therefore easily exchangeable. A user (A) is normally identified by the system after login. Possible (P) ratings (D) of objects and other users' evaluations (T) are also defined by the metadata ontology. Currently we allow a star based rating in combination with a text based explanation for the ontologies and a simple yes/no statement for the helpfulness of reviews. R and W are stored in a relational database for easier access. It is important to notice that the parameters of the Open Rating System (like allowed ratings or captures properties of ontologies) can easily be switched using a modified metadata ontology. This is the strength of having a framework that is generated dynamically from the underlying ontology.

6. PLANNED EVALUATION

We implemented and put online an alpha version of our repository with the task to gather data (such as annotations on ontologies or reviews).[16] Several ontologies have been submitted by their authors, and some of them feature reviews. We will try to increase popularity of the portal and convince people to evaluate ontologies. When we gather enough data, we will compare the performance of our approach with that of the traditional Open Rating System approach. In order to perform the comparison, we will present users with both the rankings computed based on our approach and those yielded by a traditional Open Rating System. The users will then evaluate which model provides ranking results that are more useful for him. In order to have reviews of the complete ontology for that comparison, we currently allow users to provide a "short" review of an ontology instead of the intended "detailed" review.

7. CONCLUSION

We have presented a framework for ontology evaluation consisting of an ontology repository that has already been implemented and an Open Rating System extended with topic-specific trust. The main advantage of evaluating ontologies in the context of an Open Rating System is the possibility to infer personalized rankings of ontologies based on a user's requirements and preferences. Our solution allows combining only the best evaluations for separate properties of an ontology to one global overall rating. In contrast to other approaches, ontology experts are not predefined, but democratically elected by all users, based on votes on the helpfulness of their reviews. User's can search ontologies based on needed characteristics. Reviews on the usability of an ontology can provide valuable feedback for ontology reuse.

Acknowledgments

Research reported in this paper has been partially financed by the EU in the IST projects:
NeOn (IST-2006-027595) http://www.neon-project.org
SEKT (IST-2003-506826) http://www.sekt-project.com

[13] http://tinyurl.com/qfp2s
[14] www.dmoz.org
[15] Google Directory (http://directory.google.com/) is also based on DMOZ
[16] http://smi-protege.stanford.edu:8080/KnowledgeZone

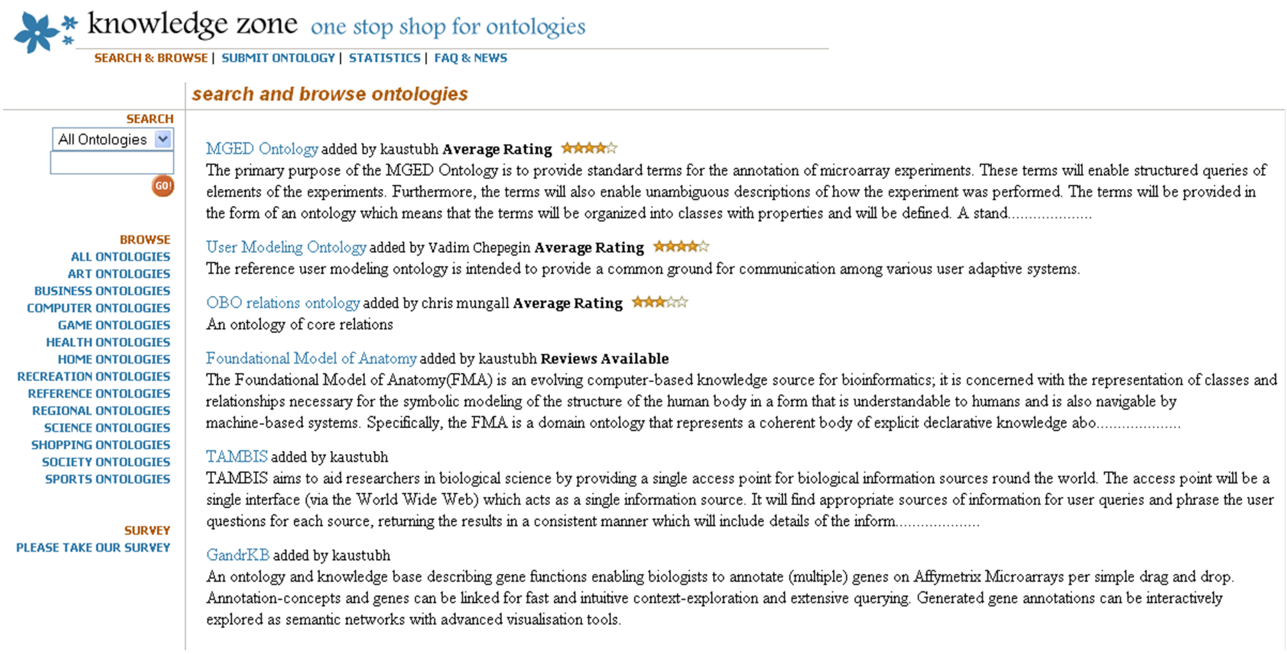

Figure 1: Entry Page of KnowledgeZone

8. REFERENCES

[1] J. Brank and M. Grobelnik. A survey of ontology evaluation techniques. In *Proc. of the 8th International mulit-conference Information Society IS*, pages 166–169, 2005.

[2] L. Ding, T. Finin, A. Joshi, R. Pan, R. S. Cost, Y. Peng, P. Reddivari, V. C. Doshi, and J. Sachs. Swoogle: A search and metadata engine for the semantic web. In *Proc. of the Thirteenth ACM Conference on Information and Knowledge Management*, pages 652–659. ACM Press, NOV 2004.

[3] J. Golbeck. *Computing and Applying Trust in Web-based Social Networks*. PhD thesis, University of Maryland. College Park, MD, 2005.

[4] A. Gómez-Pérez, M. Fernández-López, and O. Corcho. *Ontological Engineering*. Springer, London, U.K., 2003.

[5] R. Guha. Open rating systems. Technical report, Stanford University, CA, USA, 2003.

[6] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins. Propagation of trust and distrust. In *Proc. of the Thirteenth International World Wide Web Conference*, MAY 2004.

[7] J. Hartmann, Y. Sure, P. Haase, R. Palma, and M. del Carmen Surez-Figueroa. Omv – ontology metadata vocabulary. In C. Welty, editor, *ISWC 2005 - In Ontology Patterns for the Semantic Web*, NOV 2005.

[8] A. Lozano-Tello and A. Gómez-Pérez. Ontometric: A method to choose the appropriate ontology. *Journal of Database Management*, 15(2):1–18, APR-JUN 2004. Special Issue on Ontological analysis, Evaluation, and Engineering of Business Systems Analysis Methods.

[9] D. L. McGuinness. Ontologies come of age. In D. Fensel, J. Hendler, H. Lieberman, and W. Wahlster, editors, *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential*. MIT Press, 2002.

[10] N. F. Noy, R. Guha, and M. A. Musen. User ratings of ontologies: Who will rate the raters? In *Proc. of the AAAI 2005 Spring Symposium on Knowledge Collection from Volunteer Contributors*, Stanford, CA, 2005.

[11] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford University, CA, USA, 1998.

[12] C. Patel, K. Supekar, Y. Lee, and E. Park. Ontokhoj: A semantic web portal for ontology searching, ranking and classification. In *Proc. of the Fifth ACM Workshop on Web Information and Data Management*, pages 58–61, New Orleans, Louisiana, USA, 2003. ACM Press.

[13] K. Supekar and M. Musen. Knowledge zone: A public repository of peer-reviewed biomedical ontologies. *submitted to AMIA Annual Symposium*, Washington DC, USA, 11-15 November 2006.

[14] K. Supekar, N. Noy, D. Rubin, and M. Musen. Ontology of ontology to support ontology search and evaluation. *submitted to AMIA Annual Symposium*, Washington DC, USA, 11-15 November 2006.

[15] K. Supekar, C. Patel, and Y. Lee. Characterizing quality of knowledge on semantic web. In *Proc. of the Seventeenth International FLAIRS Conference*, pages 220–228, Miami, Florida, USA, MAY 2004.

[16] M. Uschold, M. Healy, K. Williamson, P. Clark, and S. Woods. Ontology reuse and application. In *Proc. of the International Conference on Formal Ontology and Information Systems FOIS'98*, pages 179–192, 1998.

[17] C. Welty and N. Guarino. Supporting ontological analysis of taxonomic relationships. *Data & Knowledge Engineering*, 39:51–74, 2001.