Extending Open Rating Systems for Ontology Ranking and Reuse

Holger Lewen[1] and Mathieu d'Aquin[2]

[1] Institute AIFB, Karlsruhe Institute of Technology (KIT), Germany
holger.lewen@kit.edu
[2] Knowledge Media Institute (KMi), The Open University, United Kingdom
m.daquin@open.ac.uk

Abstract. Ontology reuse saves costs and improves interoperability between ontologies. Knowing which ontology to reuse is difficult without having a quality assessment. We employ user ratings to determine the user-perceived quality of ontologies. The combination of an Open Rating System (ORS), user ratings, and information on trust between users, allow us to compute a personalized ranking of ontologies. In this paper, we present our extension, the Topic-Specific Trust Open Rating System (TS-ORS). To overcome the limitations of the ORS, the TS-ORS features topic-specific trust and multi-faceted ratings. In a user study, we show that having user ratings and result ranking based on a TS-ORS significantly facilitates ontology assessment and selection for the end user.

1 Introduction

Ontology reuse reduces costs and development effort [1]. Ontology engineers save time when they reuse existing ontological content—entire ontologies or parts thereof. A typical problem users face today when they try to reuse existing ontological content is the selection of the most appropriate ontology for their task. This is because the assessment of ontologies is a time-consuming and nontrivial task. So far, no automatic evaluation technique exists that can judge the quality of an ontology the way a human can. The ability to rely on the experience of other users can lessen the assessment effort considerably. Because of this we gather user-based evaluations of ontological content in our Cupboard[3] ontology publishing system [2]. Based on these evaluations we provide user-perceived quality information on the content, and, together with information on inter-user trust, we compute a user-specific (personalized) ranking of ontological content.

Our TS-ORS provides a complete ranking framework for multi-faceted ratings based on fine-grained inter-user trust and meta-trust statements. In Cupboard, we expose the ontology ratings, and provide a personalized ontology ranking based on scores computed by the TS-ORS. The main contributions of this paper are: We extend the current state of the art ORS, introduced in Section 2, to overcome two major limitations: We introduce a fine-grained topic-specific

[3] http://cupboard.open.ac.uk/

trust system including the ability to provide meta-trust, and allow multi-faceted ratings (see Section 3 for a description of our TS-ORS). We then show how the TS-ORS was adapted for ontology ranking in Cupboard (see Section 4). A user-study confirms our hypothesis that employing the TS-ORS in an ontology reuse scenario facilitates the selection and reuse of ontological content significantly compared to state of the art ontology search engines (see Section 4). Related work can be found in Section 5, followed by a conclusion in Section 6.

2 Open Rating Systems

An ORS is a system that allows users to write reviews and give ratings to arbitrary content. Other users can then trust or distrust the reviewers. Based on the trust information and the reviews collected, the system can generate a ranking for both reviews and reviewed content. The concept of ratings and meta-ratings (other users rating the ratings) has found wide adoption in the e-commerce world. A prominent example are the user reviews found on Amazon.[4]

One of the key components of the ORS model is the trust users express towards each other. The Web of Trust (WOT) these user-to-user trust statements form can be used for both *local* (user-specific) and *global* (user-agnostic) trust computation, as shown in Section 3. In order to demonstrate the differences between the original ORS model and our extension, we first present Guha's model [3], and then our extension in detail.

Model: Guha's ORS model consists of the following components:
1. A set of objects $O : \{o_1, o_2, o_3, \cdots\}$ that can be rated.
2. A set of agents $A : \{a_1, a_2, a_3, \cdots\}$ that participate in the ORS.
3. A set of possible values for ratings of objects $D : \{d_1, d_2, \cdots\}$.
4. A set of possible values for trust ratings of agents $T : \{t_1, t_2, \cdots\}$.
5. A partial function $R : A \times O \to D$ corresponding to agent ratings on objects.
6. A partial function $W : A \times A \to T$ corresponding to inter-agent trust.

Limitations of the ORS: First, there is no concept of a multi-faceted review, that means, it is only possible to provide an overall rating for an object, and not for the different aspects of an object (cf. the signature of the rating function R). In case certain aspects of an object are good, and others are bad, this derivation from the average is lost. In the context of ontology evaluation, for example, an ontology can be highly reusable, while only covering a limited part of the intended domain. In this case, reusability and domain coverage represent aspects that can be reviewed independently. Analyzing Cupboard data, we discovered that indeed many ontologies have a variance in the ratings on their different aspects. Taking one ontology space[5] we looked for cases where a reviewer provided reviews for every aspect of an ontology, allowing to compute an average rating this reviewer might have given in case only an overall rating would have been allowed. Based on 145 ratings, which would correspond to 29 overall ratings in the

[4] `http://www.amazon.com/`
[5] `http://cupboard.open.ac.uk:8081/cupboard/Experiment1`

ORS, we computed the mean variance between aspects as 0.85. This validated the extension of ORS with multi-faceted reviews.

Second, reviewers do not always write either good or bad reviews, but depending on the ontology and aspect they review, they can be trusted differently. Indeed, we observed many users that trust a review from a particular reviewer and distrust another review from that same reviewer. In the ORS model, the trust function W only stores global trust or distrust (covering all reviews of a user). To overcome this, we implement a more fine-grained trust management.

3 Topic-Specific Trust Open Rating Systems

Model: We introduce aspects of objects that can be rated, and extend the rating function R and the trust function W. In the TS-ORS model, we reuse O, A, D, T from Guha's ORS model (see Section 2) and introduce:

- A set of object aspects $X : \{x_1, x_2, x_3, \cdots\}$
- A partial function $R : A \times O \times X \to D$. R corresponds to the rating of an agent on one certain aspect of an object. Let $B_R \subseteq A \times O \times X$ denote the set of all triples for which R is defined, i.e., for which ratings exist.
- A partial function $W : A \times A \times O \times X \to T$, which corresponds to the trust of an agent in another agent for a specific aspect–object combination.

We assume all sets to be finite. In this paper we use the term *user* or *reviewer* to refer to agents from A. In our model, a reviewer has to justify each rating R with a textual review justifying the rating. We refer to the textual justification as the *review*, and to the actual D value as the *rating*. In this section, we separate the description of the model and algorithms from our concrete instantiation. The adaptation of the TS-ORS for ontology ranking alongside default values for parameters from the following algorithms can be found in Section 4.

Meta-trust Statements in the TS-ORS Model: Users express trust on the level of an aspect of an object (see definition of W). For the convenience of users, we defined and allow the use of meta-trust statements, which are trust statements covering more than one object–aspect combination. They can be seen as shortcuts to making many W statements (see Table 1).

Table 1. Allowed User-to-User Meta-trust Statements.

Statement	Signature	Explanation
W	$A \times A \times O \times X \to T$	Trust to review a specific aspect of a specific object
W_X	$A \times A \times O \to T$	Trust to review all aspects of a specific object (for arbitrary aspects of X)
W_O	$A \times A \times X \to T$	Trust to review a specific aspect of all objects (for arbitrary objects of O)
W_{OX}	$A \times A \to T$	Trust to review all aspects of all objects (for arbitrary aspects of X and objects of O)

Algorithms: In the following we show how the trust information stored in W and the meta-trust statements can be used to provide a ranking of reviews for a given aspect–object combination, and also the computation of an overall rating for objects (taking R into account). The process starts with the materialization of meta-trust to normal trust statements. We then use the trust information to compute trust values for each user, which can be used to rank reviews. Based on the top ranked reviews, we can compute an overall rating of objects.

Meta-trust Propagation: Since the meta-trust statements are not part of the model, we have to materialize them to single W statements before the trust computation. The materialization is based on our intuition that more specific trust statements (those covering a smaller scope) are more authoritative: $W \succ W_X \succ W_O \succ W_{OX}$ ("$\succ$" meaning more authoritative). The materialization is performed based on the above order, i.e., starting with all statements in the form W, then processing all statements in the form W_X, then W_O, and finally W_{OX}. For each of the meta-trust statements, their scope is checked and then the value of the statement is propagated to the object–aspect level. Existing trust information is not overwritten in this process. For example, if a meta-trust statement has been made for an object (W_X), it is checked which aspects of this object are not covered by trust statements yet, and the value of the meta-trust statement is used for these aspects. We refer to the final outcome of the materialization as W'. Using meta-trust statements, it is possible for example to distrust other users globally, but to trust them for a certain object o_n (since the statement on the object is more authoritative than the global statement, the users will be distrusted for all objects except for o_n).

Computing Trust Values for Ranking: After the meta-trust has been materialized, all statements are in the form of W, and trust ranks can be computed. Note that in contrast to Guha's ORS, we compute individual trust relationships for every aspect of every object (every $o_n x_k$ combination for $o_n \in O, x_k \in X$).[6]

For each $o_n x_k$ combination, we define two matrices storing trust and distrust. The trust matrix $\mathbf{T}$ has entries $t_{ij} \in \{0, 1\}$. If $t_{ij} = 1$, user u_i trusts u_j according to W', otherwise no information is available. Analogously, the distrust matrix $\mathbf{D}$ has entries $d_{ij} \in \{0, 1\}$ capturing the distrust. Given $\mathbf{T}$ and $\mathbf{D}$, we can compute the *GlobalTrustRank* (GTR) (a relabeled Page-Rank [4], see Equation 1), *GlobalDistrustRank* (GDR) (from [3], see Equation 2), and using Algorithm 1 (which is based on [5]) the local trust matrix $\mathbf{F}$ and the interpretation matrix $\mathbf{I}$.

$$GTR_{i+1}(u_u) = (1 - d) + d \cdot \left(\sum\nolimits_{v \in T_v} \frac{GTR_i(v)}{N_v} \right) \tag{1}$$

where u_u is the user whose GTR is computed, $v \in T_v$ is the user trusting u_u, N_v is the total number of users user v trusts, d is a damping factor between 0 and 1,[7] and i is the number of iterations the algorithm has run. Intuitively speaking, GTR assigns trust to users based on how many other users trust them and how

[6] In the following, whenever we use the subscript $o_n x_k$ for matrices or ranks, it means that they contain the data relevant to this $o_n x_k$ combination.

[7] Based on [4], it is usually set to 0.85 for fast convergence of ranks.

trusted the users trusting them are.

$$GDR(u_u) = \sum\nolimits_{v \in B_v} \frac{GTR(v)}{N_v} \tag{2}$$

where u_u is the user whose *GDR* is computed, $v \in B_v$ is the user distrusting u_u, N_v is the total number of users user v distrusts. *GDR* is taking into account who distrusts a user and how high the *GTRs* of the distrusting users are.

We base our Local Trust Computation Algorithm (see Algorithm 1) on work from Guha et al [5], who investigated trust and distrust propagation in a WOT based on real world data. The algorithm uses the 4 different kinds of atomic trust propagation shown in Table 2. It employs a combination of these trust propagation techniques to propagate trust within the WOT. Distrust is only propagated 1 step (statements from distrusted users are discounted), since the semantics of propagation for distrust are not clear [5]. After the local trust matrix **F** is computed, it is interpreted. This is done by first checking which of the resulting entries f_{ij} were originally trusted or distrusted, and initializing the interpretation matrix **I** based on that information. Then, for each row in **F** the values are ordered, and unknown interpretations for entries f_{ij} are determined by comparison to the interpretation of neighboring entries.

Ranking Reviews at the Aspect Level of an Object: When a user requests the reviews for an object–aspect combination, the reviews for that combination have to be ranked by the TS-ORS to be provided in a user-specific order. If a user is logged in, we can base the user-specific ranking on the *local* trust information. This ranking is also influenced by a parameter $\alpha \in [0, 1]$ that the user can provide to combine global trust (*GTR*) and distrust (*GDR*) to a *GlobalCombinedRank* (*GCR*). The higher α is, the more emphasis is put on the *GTR*. In case a user is logged in, we use Algorithm 2, in the other case Algorithm 3. Algorithm 2 ranks reviews based on the information who is trusted and who is distrusted, and the local trust value from **F**. In case **F** cannot be interpreted, *GCR* values are used. Algorithm 3 can only rank based on the *GCR* values, since the user is unknown. Thus its ranking cannot be personalized.

Computing an Overall Rating of an Object: Each object has aspects $x_k \in X$ it can be reviewed on. The user can choose how to weigh each aspect for computing the overall rating of an object by specifying a weight $\mu_k \geq 0$ for each aspect $x_k \in X$. In case no rating exists for an aspect, we distribute its weight to the remaining aspects. The user can also decide on how many top-ranked ratings per aspect the computation is based. This is done with the parameter $N \geq 1$. $N = 3$ means, for example, that the top-3-ranked ratings are considered. Another parameter $\nu \in (0, 1]$ can be specified that determines how the top-N-ranked ratings are combined. The closer ν is to 0, the more emphasis is put on the top-ranked ratings, if $\nu = 1$ a linear combination is computed. We use Algorithm 4 to compute the overall rating. The ranking of reviews is determined

Input: Trust Matrix $\mathbf{T}$, Distrust Matrix $\mathbf{D}$
Output: Local Tust Matrix $\mathbf{F}$, Interpretation Matrix $\mathbf{I}$

$\mathbf{C} := \beta_1 \cdot \mathbf{T} + \beta_2 \cdot \mathbf{T}^\intercal \mathbf{T} + \beta_3 \cdot \mathbf{T}^\intercal + \beta_4 \cdot \mathbf{T}\mathbf{T}^\intercal$
$\mathbf{F} := \sum_{k=1}^{K} \gamma^k \cdot (\mathbf{C}^{(k)} \cdot (\mathbf{T} - \mathbf{D}))$
Initialize $\mathbf{I}$:=$\mathbf{T} - \mathbf{D}$
foreach $j \in A$ **do**
  Compute a sequence $a(1), a(2), \ldots, a(|A|)$ such that:
  – for all $a \in A$ there is exactly one $i \in \{1, \ldots, |A|\}$ such that $a(i) = a$
  – for all $i \in \{1, \ldots, |A| - 1\}$ we find that $f_{ja(i)} \leqslant f_{ja(i+1)}$
  To simplify notation, we use $i_{ja(0)}$ and $i_{ja(|A|+1)}$ to denote 0. Small letters with subscripts denote entries in the matrices.
  repeat
    foreach $m \in A$ **do**
      if $i_{ja(m)} = 0$ **then**
        $i_{ja(m)} := 1$
        if $\Big((i_{ja(m-1)} + i_{ja(m+1)} \leq -1) \vee \big((i_{ja(m-1)} = -1 \wedge i_{ja(m+1)} = 1) \wedge ((f_{ja(m+1)} - f_{ja(m)}) > (f_{ja(m)} - f_{ja(m-1)}))\big) \vee \big((i_{ja(m-1)} = 1 \wedge i_{ja(m+1)} = -1) \wedge ((f_{ja(m+1)} - f_{ja(m)}) < (f_{ja(m)} - f_{ja(m-1)}))\big)\Big)$ **then**
          $i_{ja(m)} := -1$
        if $i_{ja(m-1)} = i_{ja(m+1)} = 0$ **then**
          $i_{ja(m)} := 0$
  until *no further changes occur in* $\boldsymbol{I}$;
  foreach m **do**
    if $f_{ja(m)} = t_{ja(m)} = d_{ja(m)} = 0$ **then**
      $i_{ja(m)} := 0$

Algorithm 1: Local Trust Computation:

by Algorithm 2 or 3, based on whether a user is logged in or not. With this extension it is possible to compose an overall rating using ratings on different aspects and based on topic-specific trust statements.

In a simulation, which can be found in our Technical Report [6], we show how TS-ORS provides better ranking results than the ORS.

4 Adaptation of the TS-ORS to Ontology Ranking

The idea of employing user ratings for ontology evaluation has first been proposed by Noy et al. [7] in 2005. In the context of ontology evaluation, the set of objects O can contain complete ontologies, parts of an ontology (modules), or even URIs of classes. In Cupboard we instantiate the aspects X based on Gangemi's work on ontology evaluation [8], using: reusability, correctness, complexity, domain coverage and modeling. We allow users to review ontologies either they or other users uploaded. We currently assume that the ratings on ontology aspects cover all axioms of an ontology. If users wish to only rate parts of an

Table 2. Atomic Trust Propagation based on [5]

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

ontology, they can do so by extracting these parts using state of the art modularization techniques, and then uploading them as a separate ontology to Cupboard. We are aware that providing highly customizable algorithms with many different aspects can confuse users, since knowing which parameter has which effect on the result is not trivial. It is important to note that many parameters have to be chosen once when instantiating the system, and thereafter are normally not changed. Moreover, providing reasonable default values allows users to interact with the system without providing their own values, while still enabling expert users to take full advantage of the flexibility of the system. In Cupboard, for T and D, we chose $T = \{trust, distrust\}$ and $D = \{1, \cdots, 5\}$, because many users are familiar with the 5 star rating schema and the possibility to assign trust and distrust (e.g., from Amazon.com). We assume the D values are equidistant. For the trust computation in Cupboard we use $\beta_1 = 0.4, \beta_2 = 0.4, \beta_3 = 0.1, \beta_4 = 0.1$, and $\gamma = 0.9$, based on an analysis of the evaluation results from [5]. We propagate trust 7 steps ($K = 7$ based on the idea of 6 degrees of separation [9]). For the user-specific part of the algorithms we use $\alpha = 0.7$, $N = 1, \nu = 0.8$ and $\mu_k = 1/|X|$ as default values, unless we have user-specified values.

In the following we show how exposing the user reviews and providing ontology ranking based on the TS-ORS can facilitate ontology reuse for the user.

Reuse User Study: The typical ontology reuse process consists of finding ontologies to reuse, then assessing and selecting them, and finally integrating

Input: $o_n \in O, x_k \in X, GTR_{o_n x_k}, GDR_{o_n x_k}, \mathbf{F}_{o_n x_k}, \mathbf{I}_{o_n x_k}, \alpha_i \in [0,1], a_i \in A$
Output: Sorted Reviews
foreach $(a_j, o_n, x_k) \in B_R$ **do**
    $GCR(a_j) := \alpha_i \cdot GTR_{o_n x_k}(a_j) - ((1 - \alpha_i) \cdot GDR_{o_n x_k}(a_j))$
    Assign to the review $(a_j, o_n, x_k) \in B_R$ a triple $(i_{ij}, f_{ij}, GCR(a_j))$
Based on these triples, reviews are sorted in descending lexicographic order (meaning columns are sorted descending, starting with i_{ij} and then considering f_{ij} to sort entries where the i_{ij} value is identical, and then considering $GCR(a_j)$ if both values for i_{ij} and f_{ij} are identical)

Algorithm 2: Review Ranking if the User Can be Identified

Input: $o_n \in O, x_k \in X, GTR_{o_n x_k}, GDR_{o_n x_k}, \alpha \in [0,1]$
Output: Sorted Reviews
foreach $(a_j, o_n, x_k) \in B_R$ **do**
    $GCR(a_j) := \alpha \cdot GTR_{o_n x_k}(a_j) - ((1-\alpha) \cdot GDR_{o_n x_k}, (a_j))$
    Assign to the review $(a_j, o_n, x_k) \in B_R$ a value $GCR(a_j)$
Based on these values, reviews are sorted starting with the highest value.

Algorithm 3: Review Ranking if the User Cannot be Identified

them. Up to now, ontology search engines do not provide help for ontology selection to users searching for ontological content to reuse. Since rating systems can be used to facilitate exactly this step in the reuse process, we ran a user study comparing the helpfulness of our TS-ORS compared to ontology search engines available on the Web and against Watson. For the experiment, we extended the Watson plug-in [12] for the NeOn Toolkit[8] to get the user evaluations from Cupboard and base the result ranking on the overall ratings as computed by Algorithm 4. We refer to this plug-in as the Cupboard plug-in. The experiment had 20 participants from 6 different institutions (17 PhD students, two postdocs and one professor). We assigned each participant to one of three groups. The task for each group was to extend a given ontology reusing existing ontological content. All groups had access to online ontology search engines. Group 1 had no additional plug-ins to facilitate the reuse process. Group 2 had the Watson plug-in at their disposal, and Group 3 could use the Cupboard plug-in. All participants filled out a questionnaire. The detailed experiment description and analysis of

[8] `http://neon-toolkit.org/`

Input: $o_n \in O$, ranked sets of reviews
    $B_{o_n x_k} = \{(a_{j1}, o_n, x_k), \cdots, (a_{jm}, o_n, x_k)\} \subseteq B_R$ for each $x_k \in X$,
    $\nu \in (0,1]$, $N \geqslant 1$, μ_k for each $x_k \in X$
Output: Rating D_{o_n}
For a given $o_n x_k$ combination, we use the notation $(a_{ji}, o_n, x_k) \in B_{o_n x_k}$ to refer to the i-th ranked result.
foreach $x_k \in X$ **do**
    $N := \min(N, |B_{o_n x_k}|)$
    if $N = 0$ **then**
        $D_{o_n x_k} := 0$
        $\mu_k := 0$
    else
        $D_{o_n x_k} := \frac{1}{N} \sum_{i=1}^{N} \left((\nu^i / \sum_{s=1}^{N} \nu^s) \cdot R(a_{ji}, o_n, x_k) \right)$
foreach $k = 1, \cdots, |X|$ *with* $\mu_k \neq 0$ **do**
    $\mu_k := \mu_k / \sum_{l=1}^{|X|} \mu_l$
$D_{o_n} := \sum_{x_k \in X} \mu_k \cdot D_{o_n x_k}$

Algorithm 4: Computation of an Overall Rating

Table 3. Partial questionnaire results including two-sided p values for pairwise group comparison based on Fisher's exact test [10] with Yate's continuity correction [11].

Question: Did you have trouble finding ontology statements to reuse?						
	Group 1	Group 2	Group 3	**p** Gr. 1 vs Gr 2.	**p** Gr. 1 vs Gr. 3	**p** Gr. 2 vs Gr. 3
Yes	6	1	1	0.0047	0.0047	1
No	0	6	6			
Question: Did you have trouble selecting ontology statements to reuse?						
	Group 1	Group 2	Group 3	**p** Gr. 1 vs Gr 2.	**p** Gr. 1 vs Gr. 3	**p** Gr. 2 vs Gr. 3
Yes	5	5	0	1	0.0047	0.021
No	1	2	7			
Question: Did you have trouble integrating ontology statements to reuse?						
	Group 1	Group 2	Group 3	**p** Gr. 1 vs Gr 2.	**p** Gr. 1 vs Gr. 3	**p** Gr. 2 vs Gr. 3
Yes	5	1	0	0.0291	0.0047	1
No	1	6	7			

both the resulting ontologies and the questionnaire results can be found in [13]. We focus here on the key findings with regard to the three challenges of finding, selecting and integrating reusable ontological content. Table 3 provides the three most important questions covering these three challenges. As is evident from the table, users using either the Watson or the Cupboard plug-in had no trouble finding or integrating ontological content, compared to users who only could use ontology search engines on the Internet (based on p values, the findings are highly significant for $p = 0.0047$ and significant for $p=0.0291$). Group 3, which had the results ranked based on user ratings, stated they had no problem selecting ontology statements compared to both the Watson group (whose result ranking was based on Lucene), or group 1 which was using a plethora of Semantic Web search engines on the Internet. Again, this result is statistically significant and shows that users have less problems selecting content when offered TS-ORS ranking and scores compared to the current state of the art.

5 Related Work

Our closest related work is Guha's work on Open Rating Systems [3], and trust propagation [5]. To overcome limitations of the related work, we have extended the model with ratings on aspects of objects, and provided a comprehensive framework for fine grained topic-specific trust and meta-trust expression. In contrast to the related work, we present a full algorithmic description and complete framework for the computation of personalized ratings based on ratings on objects and fine-grained user trust.

6 Conclusion

We have presented the TS-ORS which features multi-aspect object reviews and topic-specfic trust. The user study based on our implementation inside Cupboard has shown that users have significantly fewer problems selecting ontological content to reuse when provided user-based information on the quality of the ontologies, as delivered by our TS-ORS, compared to other state of the art ontology search engines.

Acknowledgements

Research reported in this paper was partially funded by the European Commission through the IST project ACTIVE (ICT-FP7-215040). We thank Markus Krötsch, Denny Vrandecic, Elena Simperl, Andreas Harth, Thanh Tran, Barry Norton, Sudhir Agarwal, and Natasha Noy for their valuable feedback.

References

1. Simperl, E.P.B., Popov, I.O., Bürger, T.: Ontocom revisited: Towards accurate cost predictions for ontology development projects. In: Proc. of the 6th ESWC (ESWC 2009). Volume 5554 of LNCS., Springer (MAY 2009) 248–262
2. d'Aquin, M., Lewen, H.: Cupboard – A Place to Expose your Ontologies to Applications and the Community. In: Proc. of the 6th European Semantic Web Conference (ESWC 2009). Volume 5554 of LNCS., Springer (MAY 2009) 913–918
3. Guha, R.: Open rating systems. In: 1st Workshop on Friend of a Friend, Social Networking and the Semantic Web. (2004)
4. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web. Technical report, Stanford University, CA, USA (1998)
5. Guha, R., Kumar, R., Raghavan, P., Tomkins, A.: Propagation of trust and distrust. In: Proc. of the Thirteenth International World Wide Web Conference, New York, NY, ACM Press (MAY 2004) 403–412
6. Lewen, H.: Simulation-based Evaluation of the Topic-Specific Trust Open Rating System. Technical report, Universität Karlsruhe (TH) (JUN 2009) `http://www.aifb.uni-karlsruhe.de/WBS/hle/paper/TR2.pdf`.
7. Noy, N., Guha, R., Musen, M.: User ratings of ontologies: Who will rate the raters. In: Proceedings of the AAAI 2005 Spring Symposium on Knowledge Collection from Volunteer Contributors. (2005)
8. Gangemi, A., Catenacci, C., Ciaramita, M., Lehmann, J.: Modelling ontology evaluation and validation. In Sure, Y., Domingue, J., eds.: ESWC. Volume 4011 of Lecture Notes in Computer Science., Springer (2006) 140–154
9. Milgram, S.: The small world problem. Psychology today **2**(1) (1967) 60–67
10. Fisher, R.: On the interpretation of χ 2 from contingency tables, and the calculation of P. Journal of the Royal Statistical Society **85**(1) (1922) 87–94
11. Yates, F.: Contingency tables involving small numbers and the χ 2 test. Supplement to the Journal of the Royal Statistical Society (1934) 217–235
12. d'Aquin, M., Motta, E., Dzbor, M., Gridinoc, L., Heath, T., Sabou, M.: Collaborative semantic authoring. IEEE IS, IEEE **23**(3) (May-June 2008) 80–83
13. Lewen, H.: Facilitating Ontology Reuse with a Topic-Specific Trust Open Rating System. Technical report, Universität Karlsruhe (TH) (JUN 2009) `http://www.aifb.uni-karlsruhe.de/WBS/hle/paper/TR3.pdf`.