

# TRUST CALCULATION

## *Semantic Agreement for Ontology Integration*

Dennis Hooijmaijers

*ACRC, University of SA, Australia*

dennis.hooijmaijers@unisa.edu.au

Markus Stumptner

*ACRC, University of SA, Australia*

mst@cs.unisa.edu.au

**Abstract** The Semantic Web is generally envisioned as a vast collection of document embedded knowledge that makes it highly improbable for agents traversing this space to know directly what entity, person, or organisation they are dealing with. In such an environment, the explicit representation of trust becomes an intrinsic part of calculating whether an agent can believe and use (or reuse) foreign sources. A key activity in this process is the step of integrating an agent's ontology with that of another document found on the Web. To assist in calculating trust values for this purpose, Riposte provides a set of trust models and trust manipulation algorithms to create a dynamic model of author trust based on work that is being provided to an agent. Riposte is an ontology integration tool that uses suggestions and bases trust on whether an object in the provided ontology confirms or refutes current beliefs. The author can be assigned an initial trust value and this value is recalculated after the integration process.

**Keywords:** Ontology, Bayesian Network, Trust

## 1. INTRODUCTION

Authors can often refute or confirm claims of another author, and this can cause difficulties when merging ontologies, ensuring inconsistencies are not introduced and usability disrupted (Hooijmaijers and Bright, 2005). When considering an ontology for integration with one's existing conceptual setting, it is clearly desirable to ensure the reliability and credibility of the incoming ontology. A key consideration is the reliability and credibility of its author. Current integration solutions generally presume ontologies to be certain, error free, and take no consideration for the differences in bias. No other application scenario challenges this view more than the Semantic Web.

The Semantic Web will be a vast network that makes direct knowledge of authors unattainable (Ziegler and Lausen, 2004) and will often require an agent to outsource its requests for knowledge (Huang and Fox, 2005), resulting in knowledge being passed on from unknown agents, and being integrated with the requesting agent's ontology. In such cases it is necessary to ensure that such passed-on knowledge is credible and that the agent is reliable, and provisions for coherently integrating multiple, possibly mutually contradicting, sources, must exist. Any errors or inconsistencies provided by the new agent may cause the entire ontology to become unusable.

This paper introduces Riposte, an approach to ontology integration that takes into consideration the uncertainties, contained within the ontologies, and the trust and credibility of the author. Riposte attempts to minimise the harm caused by unknown authors by providing trust values for each component of the provided ontology, allowing for rating the providers based on the work they provide, thus creating a local trust metric (Ziegler and Lausen, 2004). Riposte provides an annotation model, a belief update model and reasoner for OWL ontologies. By providing annotation to existing ontologies that capture credibility at concept and property level, Riposte can provide a more realistic approach to decision making based on multiple sources. This paper defines appropriate models for capturing, updating, and propagating uncertainty in ontologies for integration purposes.

## 2. TRUST AND ONTOLOGIES

Numerous trust models (Ziegler and Lausen, 2004; Golbeck et al., 2003; Bertino et al., 2004; Staab et al., 2005; Noy et al., 2005) have been proposed in recent years to rate or quantify belief in information provided by a person. A key distinction is between direct trust (local trust) and reputation (global trust) (Ziegler and Lausen, 2004). Direct trust is based on previous experience with the source of the information, while reputation requires a social network model to provide a similar measure. Some trust researchers (Bertino et al., 2004) also aim to provide a metric for authenticity. This 'security trust' aims to prevent document tampering (encryption) and ascertain authorship (digital signatures). These metrics often ignore the source's credibility and reliability (Golbeck et al., 2003). Credibility is the concept of trusting a source, in a domain, to provide information that is most likely to be correct (Ding et al., 2003; Golbeck et al., 2003), while reliability is the concept of ranking sources by their credibility ratings (Golbeck et al., 2003; Huang and Fox, 2005). This work focuses on the credibility of an ontology and author, and trust is considered a value that represents the belief that the author produces credible work.

In a Semantic Web environment it is unlikely that an agent will have previous experience with all agents it may need to interact with. To overcome this lack of previous experience a social model (Dumbill, 2002; Guha, 2003) is required. According to (Staab et al., 2005), communities will create their own ontologies for the Semantic Web where the social model of the commu-

nity becomes an integral part of how the ontology is created. This results in the Semantic Web consisting of communities, their ontologies, and the content expressed through them. The social models generally provide a mechanism for assigning a rating value to each person that the agent had direct contact with. These models usually assign trust for a whole topic area.

The 'Web of Trust' approach (Guha et al., 2004) provides a technique to model trust between objects (ontologies or knowledge bases), users, reviews, ratings of reviews, and trust relations between users. The model allows for trust to be captured and rated between users. Also the model allows for each review to be ranked and to allow for the propagation of trust changes. When a user's trust rank is altered, their reviews of other users will gain additional value within the model to reflect their improved reputation. One problem is that the approach does not take into account the reliance between objects. If an ontology,  $\omega_1$ , is built incorporating another ontology,  $\omega_2$ , then the rating of  $\omega_2$  can not be higher than  $\omega_1$ . Friend of a Friend (FOAF) (Dumbill, 2002) is an approach aiming to mark up trust relations between people in XML and RDF to enrich personal web pages. It provides a simple XML schema to allow an author to define people they know. This creates a directed graph with nodes representing people and edges representing direct trust. The FOAF approach has been extended to provide levels of trust, ranking from 1 'distrusts absolutely' to 9 'trusts absolutely' (Golbeck et al., 2003), and also allowing for the trust ratings to be related to specific topics. Trust can be calculated in three ways, via path capacity, path length and weighted averages. A similar approach is followed in (Ziegler and Lausen, 2004), but does not model distrust but rather 'lack of trust'.

## 2.1 Ontology Integration

Automated knowledge integration has been an active research area for some time (McGuinness et al., 2000; Noy and Musen, 1999), but so far has mostly concentrated on knowledge assumed to be stable, certain, and (for a particular problem or domain) complete. Automated advisors have been suggested to overcome the 'knowledge acquisition bottleneck' (Gonzalez and Dankel, 1993) caused by reliance on human experts. Such systems, including PROMPT (Noy and Musen, 1999), OMEN (Noy et al., 2004) and Chimaera (McGuinness et al., 2000), use simple heuristics to assist users in making the best decisions. In real world situations, of course, knowledge is subjective and the creators of the knowledge are not always completely certain of the correct semantics between the concepts. The Chimaera ontology integration tool (McGuinness et al., 2000) uses a suggestion algorithm based on pattern matching on concept labels. It searches through the first model and then, using a set of reasoning rules, searches through the second model for a match. Chimaera mainly focuses on the subsumption hierarchies within ontologies. What Chimaera attempts to do is to integrate two hierarchies contained within multiple ontolo-

gies. This is achieved by determining if two similar concepts are the same or one is a subclass of the other.

PROMPT (Noy and Musen, 1999) was developed to semi-automatically merge ontologies, by guiding the user where it is unable to decide upon the correct results. Automated identification of likely concept mappings will reduce a domain expert's effort in searching through the ontologies. OMEN (Noy et al., 2004) is an extension of PROMPT that uses Bayesian networks to assign belief values to likely matches, thereby providing a finer gradation of candidate matchings. When a user selects a match, the beliefs are updated by propagating the new evidence through the network.

These systems assume that the ontologies to be matched are accurate and contain only definite concepts and semantics. Yet, since the reason to perform an ontology matching process is presumably the need incorporate from all sources involved, any long term usage scenario will find agents possessing and working with ontologies that have already undergone this matching process. Instead of going back to the "clean" source ontologies (which in any case may have long changed themselves, requiring multiple realignment processes), the most likely scenario will be to use the trust-annotated ontologies themselves as the source.

### **3. RIPOSTE MODEL**

The Riposte approach uses a trust model and integration model. The trust model deals with trust annotation and trust manipulation, while the integration model addresses ontology management, ontology integration and reasoning issues. In this paper we discuss the trust model and ontology integration section of the integration model.

#### **3.1 Trust Annotation**

The extension of ontology languages for uncertain knowledge has been a research topic for some time. In (Ding and Peng, 2004), the Web Ontology Language OWL was extended by an additional node type that expresses probabilistic information about concepts, to allow for "more accurate semantic integration" (Ding and Peng, 2004). This approach will work in uncertain domains where the concepts allow for uncertainty in regards to data membership, but does not allow assigning probabilities to semantic relationships; uncertainty nodes can be linked only to concepts (Hooijmaijers and Bright, 2005). The approach permits the use of Bayesian Network frameworks to perform the necessary actions, like propagating and updating beliefs in the network. Earlier work by (Jaeger, 1994) also provided a mechanism for capturing probability within Terminological Logics, annotating concepts and objects with probabilities. For semantic matching and integration purposes, these capabilities need to be extended (Hooijmaijers and Bright, 2005).

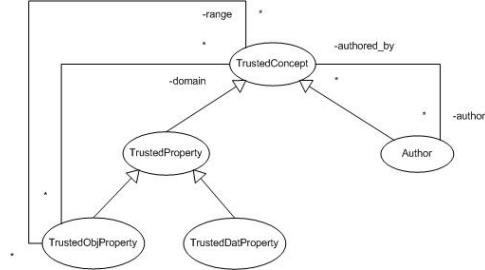


Figure 1. Necessary Concepts for Trust Annotation

Figure 1 shows the necessary concepts required for the trusted integration approached discussed later in this paper. The annotation process adds these concepts to an ontology and adds all subclasses of "owl:Thing" as subclasses of "TrustedConcept". The superclass "owl:Thing" is then removed from each of these classes. All data and object properties in the ontology are added as instances of the appropriate "TrustedProperty". Finally the author is added in as an instance of Author. Once this annotation process is complete, the ontology is ready for the integration process. The "trustvalue" is then calculated from the authors that contribute the object to the ontology.

### 3.2 The Web of Trust Model

For ontology integration it is necessary to trust that an author has every component of their ontology correct. To do, we calculate a trust value for that author for each component in their ontology, following the "Web of Trust" (Guha et al., 2004) model. This approach provides an ability to apply trust to each individual object provided by an agent, thus providing a finer grained approach to ontology trust than just evaluating topics as a whole, as proposed by (Golbeck et al., 2003). The 'Web of Trust' model consists of:

- 1 A Set of objects,  $O: \{O_1, O_2, \dots\}$ , in this case classes,  $C \subset O$ , object properties, that make up the set of objects.
- 2 A set of agents,  $A: \{A_1, A_2, \dots\}$ , who are authors that in this case contribute a class or property, by using it in their ontology.
- 3 A set of possible values,  $P(O): \{P(O_1), P(O_2), \dots\}$ , for rating the objects in this case values between -1 and 1, where -1 is complete distrust, 0 is unknown, and 1 is complete trust.
- 4 A set of values,  $P(A) : \{ P(A_1), P(A_2), \dots \}$ , for rating the authors.
- 5 A partial function  $R: A \times O \rightarrow P(O)$  that corresponds to agent ratings for various objects
- 6 A partial function  $W: A \times A \rightarrow P(A)$  That corresponds to agent ratings for various agents.

W becomes a directed graph similar to FOAF (Dumbill, 2002). The graph W is referred to as the 'Web of Trust'.

**Revisable Web of Trust.** Riposte extends the 'Web of Trust' to provide the ability to capture relationships between objects. The aim is to provide a mechanism that allows for trust updating on an object to effect all other objects that are related. The objects in Riposte represent the components of the ontologies as well as the authors who provide these components, enabling us to capture the relationships and functions for updating trust for individual objects and related objects.

A *revisable Web of Trust* is defined by the model above, with these additions:

- 1 an ontology  $\omega$  comprised of the set of objects, O
- 2 A partial function  $L: O \times O \rightarrow \{(O_a, O_l), (O_b, O_m), \dots\}$  that corresponds to a set of reliance measurements between objects. For example, an object property would rely on all domain and range classes existing for it to exist. The necessity of this reliance between objects in ontologies creates the necessity to include trust manipulation.
- 3 A function  $\tau_O$ : Providing trust revision on an Object O.

### 3.3 Trust Manipulation

A number of propagation methods have been suggested as models for trust calculation (Golbeck et al., 2003; Guha et al., 2004; Ziegler and Lausen, 2004), generally trying to capture the structure of social networks that spread and communicate trust values. Specific techniques, such as FOAF, attempt to calculate the "degrees of separation" and assign values based on closeness measures (Golbeck et al., 2003). Golbeck also introduces 9 categories of trust to help differentiate between authors. The trust levels are also applied to individual topics. Other techniques use probability (Hooijmaijers and Bright, 2005) to represent trust as a level of uncertainty in the work provided by the author. Each technique has advantages; humans often find it easier to select a natural language level over numerical values, while numerical values offer a greater level of precision for use in belief revision.

**Trust Update:** Our trust update model is based on earlier work (Hooijmaijers and Bright, 2005) that used the Stanford certainty factor algebra (SCF) (Luger, 2002) for belief revision. The beliefs can be strengthened or weakened. In Riposte the trust model provides a similar belief concept, in that it is stored in the ontology in a similar way and requires belief updating. The SCF is not usable for object belief values in Riposte, as there are dependencies between the objects. This is true in particular for object properties being reliant on the classes in the domain and range of the property. Bayes Theorem is used for updating trust for each object within the integration process. In initial work the calculation of initial author trust,  $P(A)$ , is assumed to be independent. This

allows for the combination of author trusts:

$$P(A_1, A_2, \dots, A_n) = P(A_1) \cdot P(A_2) \dots P(A_n) \quad (1)$$

If an author is correct then all objects provided by that author are true  $P(O|A) = 1$ . If an object property,  $O_k$  is true this then increases the likelihood of all classes in the range  $R : \{O_1, O_2, \dots\}$ , and domain,  $D : \{O'_1, O'_2, \dots\}$  adding the links from  $O_k$  to each  $O \in R$  and  $O \in D$  to the Bayesian network. Then for the purpose of calculating ( 3),  $O_k$  is treated as a contributing Author. This is due to the fact that both  $O_k$  and the contributing authors are treated as evidence for the purpose of calculating  $P(A \rightarrow O)$ . When calculating Author trust  $P(A)$  using ( 4),  $O_k$  is not considered as a contributing author.

$$P(O) = \sum_{i=0}^n P(O) + P(A_i) - P(O) \cdot P(A_i) \quad (2)$$

As authors confirm a given object, the trust that the object is correct is increased. If an author disagrees with the object  $O_n$ , as discussed in the Riposte algorithm it is necessary to weaken the trust in that object,  $P(O_n)$ . This is equivalent to strengthening  $P(\neg O_n)$  using ( 2). Using ( 1) to calculate  $P(O)$  for multiple contributing authors Bayes Theorem gives:

$$P(O|A_1 \& A_2 \& \dots A_n) = \frac{P(O) \cdot P(A_1 \& A_2 \& \dots A_n | O)}{P(A_1 \& A_2 \& \dots A_n)}$$

Assume at least one author is correct then  $P(A_1 \& A_2 \& \dots A_n | O)$  simplifies to 1 and this gives:

$$P(A_1 \& A_2 \& \dots A_n | O) = \frac{P(O)}{P(A_1 \& A_2 \& \dots A_n)} \quad (3)$$

### 3.4 Author Trust

Author trust can be calculated in multiple ways. In this work the trust of an author is based on the trust ratings of each object provided by that author. Similar to (Ziegler and Lausen, 2004) this is a local trust metric and is a trust according to the agent that has integrated objects supplied by the author. The author trust is calculated by calculating the trust using Jeffrey's conditioning:

$$Q(A_k) = \sum_{j=0}^M P(O_j) \cdot P_{O_j}(A_k) + (1 - P(O_j)) \cdot P_{\neg O_j}(A_k) \quad (4)$$

### 3.5 Integration using PROMPT

The PROMPT algorithm, described in (Noy and Musen, 1999), is a suggestion based algorithm for integration. PROMPT uses a recursive algorithm to highlight likely mappings between the ontologies. It goes through the ontologies and finds initial suggestions. Once the user selects an operation, PROMPT updates the resultant ontology and searches for conflicts. If conflicts are found the user must resolve the conflict. Finally PROMPT checks for new possible mappings based on the operation and present the user with a new set of suggestions. Riposte extends PROMPT to provide mechanisms to annotate, manipulate, and calculate trust values. Riposte requires two ontologies  $\omega_1$  and  $\omega_2$  each containing a set of objects,  $O$ , comprised of:

- 1 Set of Object Properties  $P_o \{P_{o1}, P_{o2}, \dots\}$
- 2 Set of Data Properties  $P_d \{P_{d1}, P_{d2}, \dots\}$
- 3 Set of concepts  $C \{C_1, C_2, \dots\}$

We assume that there exist at least 2 pairs of  $C$  in both  $\omega_1$  and  $\omega_2$ . The ontologies are assumed to have the concept 'owl:Thing' in common. (Clearly, if there is no common node in  $\omega_1$  and  $\omega_2$  there is no reference for integration; also owl:Thing is the root of OWL based modeling.)

### 3.6 Integration Algorithm

The items in the following algorithm emphasised in italics are modified from the PROMPT algorithm as described in (Noy and Musen, 1999).

- 1 **If** necessary **then** annotate. This involves inserting the concepts and creating individuals as discussed in the Trust Annotation section previously. Initially all trust values of the 'TrustedConcepts' are set to the initial author trust value, using ( 2).
- 2 *Load two ontologies,  $\omega_1$  and  $\omega_2$  and select a preferred ontology. In Riposte the preferred ontology is usually that belonging to the agent performing the merge.*
- 3 *Generate the initial list of suggestions,  $S: \{S_1, S_2, \dots\}$ .*
- 4 **Begin Integration loop**
- 5 *The user selects an operation to perform. The set of possible basic operations is defined in (Noy and Musen, 1999) on an object,  $O_n$ .*
- 6 Ensure  $O_n$  has a trust value,  $P(O)$ , assigned and a set of Authors,  $A_O$ , that contributed to that trust value.
- 7 Check  $\omega_1$  for author,  $A_n \in A$ . Often a user may have previous experience with an author. In this situation  $P(O) \in \omega_2$  is calculated using ( 2). Otherwise,  $P(O) \in \omega_2$  is set to 0.
  - (a) **If** operation is a merge, **then**, once an object,  $O_n$  has been merged the object authors,  $A_{O\omega_2}$  are added to the list of authors of object,  $O_n: A_{\omega_1}$ .



- (b) Create new suggestions  $S_2$ . Add any conflict  $\gamma_i$  to the set of conflicts,  $\gamma: \{\gamma_1, \gamma_2, \dots\}$ .
  - (c) Check for refutation,  $\chi: \{\chi_1, \chi_2, \dots\}$ , and confirmation,  $\psi: \{\psi_1, \psi_2, \dots\}$ , created by operation on  $\omega_1$  and  $O_n$  refutations and confirmations are populated using the following rules:
    - i if  $O_n$  is a property, differences in Range will add to  $\psi$ .  $O_n \in \omega_1$  has a range of C  $O_n \in \omega_2$  has range of  $C'$  where  $\forall C \in C \cap C'$  are added to  $\psi$  while  $\forall C \in \overline{C \cap C'}$  are added to  $\chi$ .
    - ii if  $O_n$  is a property difference in restrictions can add to refutation. If a restriction, R, on  $O_n \in \omega_1$  contradicts, R' on  $O_n \in \omega_2$  then add  $O_n$  to  $\psi$ .
  - (d) **For each**  $O \in \chi$  update the trust in O, P(O) using ( 2).
  - (e) **For each**  $O \in \psi$  trust in O, P( $\neg O$ ) using ( 2).
  - (f) **If** operation is a copy **then** O does not exist in  $\omega_1$ ; copy all contributing authors for O to  $\omega_1$  and calculate P(O) using ( 2).
- 8 Perform automatic updates as described in (Noy and Musen, 1999).
- 9 **Repeat from 'Begin Integration loop' until the ontologies are fully merged.** Once fully merged there should be no conflicts,  $\gamma: \{\emptyset\}$ .
- 10 Calculate Author Trust. Use ( 4) for all trust values, P(O) of  $O \in \omega_1$  that has author as a contributing author  $A_i \in A$ . assign result to  $P(A)_i$

## 4. EVALUATION

We have compared Riposte to the PROMPT integration algorithm to check the resultant ontology, and to FOAF to compare the trust values of authors.

### 4.1 PROMPT Comparison

Riposte uses the same suggestion algorithm as PROMPT. To compare the results the PROMPT tutorial ontologies, an airline reservation ontology and a car rental ontology, were integrated using both Riposte and PROMPT. The steps to perform the merge were followed from the PROMPT tutorial. Once the resultant ontology was created it was necessary to remove the Riposte annotations. This is achieved by firstly changing all subclasses of 'TrustedConcept' to subclasses of 'owl:Thing'. Secondly removing 'TrustedConcept', 'Author', 'Property', 'ObjectProperty', and 'DataProperty'. Finally all individuals of these classes are also removed. The initial merging of the additional Riposte concepts annotated on the ontologies is performed automatically without the user being given the suggestions. This ensures that the Riposte suggestion list matches those given by PROMPT. By following the same order of operations, there was no visible difference between the suggestions. The resulting ontology in Riposte included all of the Riposte annotation classes. By removing the the Riposte annotations, the resultant ontologies had no were identical to the PROMPT ontologies. This result shows that Riposte has the ability to perform the same merges as PROMPT on OWL ontologies.

## 4.2 Trust Comparison

For comparison against the FOAF trust model each of the 9 levels of Golbeck's trust were assigned numerical equivalents (*1to1*). Two path length values were chosen for initial comparisons, firstly path length 1, a friend  $A_1$ , and secondly path length 2, a friend of a friend,  $A_2$ . To compare the Riposte trust value of  $A_2$ ,  $P(A_2)$ , it is necessary for  $\omega_1$  to exclude the individual  $A_2$ , while  $\omega_2$  must contain the individual  $A_2$  and some object  $A_2$  has contributed to. The initial FOAF values for  $P(A_1)$  was 0.75, while  $P(A_2)$  was assigned 0.5. Once the ontologies were set up for Riposte, two different comparisons were performed. Firstly the initial trust value of the authors contained in  $\omega_2$  were set to the FOAF values for those authors. Secondly the initial trust values of those authors were set to 0. This allowed for the comparison of Riposte to update author trusts and also to provide an initial trust value given that the authors are completely unknown. The results provided an accurate model of author trust. For two ontologies with  $A_1$  providing objects that confirmed and refuted in the following ratios, confirmation of 0.5, refutation of 0.15 and 0.35 new objects the value calculated increased by  $\sim 0.05$ . This displays an increased trust in the author, as they provided a greater percentile of confirmations. A similar result was achieved for  $A_2$  where the overall trust went down  $\sim 0.01$  due to higher refutations. The advantage over FOAF is that the ability to see exactly where the refutation appeared allows for a pinpointing of trust and distrust to a specific area of the ontology.

## 5. CONCLUSION

We have extended PROMPT ontology integration algorithm to introduce trust calculations for all concepts and properties within two ontologies being merged. To achieve this it was necessary to annotate ontologies with a set of Riposte specific classes and to propagate individuals. This has allowed for the calculation of author trust values from the knowledge sources provided for integration. In initial tests, we have found that for moderate sized ontologies the trust value of an author is almost independent of any initial values. This allows for the calculation of trust based almost entirely on the ontology provided. Our next steps for the Riposte framework are to expand the belief propagation mechanisms, to propagate belief across related objects in an ontology, and to expand the integration mechanism to incorporate prior trust values for the objects in the source ontologies. This will permit the use of uncertain sources (that may have differing trust levels associated with their parts) as part of the merging process, enabling long-term knowledge use and reuse rather than starting from scratch.

## References

- Bertino, E., Ferrari, E., and Squicciarini, A. (2004). Trust negotiations: concepts, systems, and languages. *Computing in Science and Engineering*, 06(4):27–34.
- Ding, L., Zhou, L., and Finin, T. (2003). Trust based knowledge outsourcing for Semantic Web agents. In *Proc. IEEE/WIC*, pages 379–387.
- Ding, Zhongli and Peng, Yun (2004). A probabilistic extension to ontology language OWL. In *Proc. of the Hawaii International Conference on System Sciences.*, pages 1775–1784.
- Dou, Dejing, McDermott, Drew, and Qi, Peishen. (2003). Ontology translation on the Semantic Web. In *Proc. ODBASE'03*, pages 952–969.
- Dumbill, Ed (2002). XML watch: Finding friends with XML and RDF. *IBM's XML Watch*.
- Golbeck, Jennifer, Parsia, Bijan, and Hendler, James (2003). Trust networks on the Semantic Web. In *7th International Workshop, CIA 2003*, LNAI 2782, pages 238–249.
- Gonzalez, Avelino J. and Dankel, Douglas D. (1993). *The engineering of knowledge-based systems : theory and practice*. Prentice Hall.
- Guha, R. V. (2003). Open rating systems. Technical report, Stanford Knowledge System Laboratory.
- Guha, R. V., Kumar, R., Raghavan, P., and Tomkins, A. (2004). Propagation of trust and distrust. In *Proc. WWW*.
- Hooijmaijers, Dennis and Bright, Damien (2005). Uncertain knowledge gathering: an evolutionary approach. In *Proc. ISMIS 05*, Saratoga Springs NY. Springer-Verlag.
- Huang, Jingwei and Fox, M.S. (2005). Trust judgment in knowledge provenance. In *Proc. DEXA, 2005*, pages 524–528.
- Jaeger, Manfred (1994). Probabilistic reasoning in Terminological Logics. In *Proc. KR Conf.*.
- Luger, G. F. (2002). *Artificial Intelligence Structures and Strategies for Complex Problem Solving*. Addison Wesley, 4th edition.
- McGuinness, D.L., Fikes, R., Rice, J., and Wilder, S. (2000). The Chimaera ontology environment. In *Proc. AAAI*.
- Noy, N. F., Guha, R. V., and Musen, M. A. (2005). User ratings of ontologies: Who will rate the raters? In *Proc. AAAI*.
- Noy, N. F., Mitra, P., and Jaiswal, A. R. (2004). OMEN: A probabilistic ontology mapping tool. In *Proc. ISWC '04*.
- Noy, N. F. and Musen, M. A. (1999). An algorithm for merging and aligning ontologies: Automation and tool support. In *Proc. AAAI*.
- Pearl, Judea (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann.
- Staab, S., Domingos, P., Mike, P., Golbeck, J., Ding, Li, Finin, T., Joshi, A., Nowak, A., and Vallacher, R.R. (2005). Social networks applied. *IEEE Intelligent Systems*, 20(1):80–93.
- Ziegler, Cai-Nicolas and Lausen, Georg (2004). Spreading activation models for trust propagation. In *Proc. EEE 2004*.