The SWAP Data and Metadata Model for Semantics-Based Peer-to-Peer Systems

Marc Ehrig[2], Peter Haase[2], Ronny Siebes[1], Steffen Staab[2], Heiner Stuckenschmidt[1], Rudi Studer[2], Christoph Tempich[2]

[1]Vrije Universiteit Amsterdam
{ronny, heiner}@cs.vu.nl
[2]Institute AIFB, University of Karlsruhe, D-76128 Karlsruhe
{ehrig,haase,staab,studer,tempich}@aifb.uni-karlsruhe.de

Abstract. Peer-to-Peer systems are a new paradigm for information sharing and some systems have successfully been deployed. It has been argued that current Peer-to-Peer systems suffer from the lack of semantics. The SWAP project (Semantic Web and Peer-to-Peer) [1] aims at overcoming this problem by combining the Peer-to-Peer paradigm with Semantic Web technologies. In this paper, we propose a data model for encoding semantic information that combines features of ontologies (concept hierarchies, relational structures) with a flexible description and rating model that allows us to handle heterogeneous and even contradictory views on the domain of interest. We discuss the role of this model in the SWAP environment and describe the model as well as its application.

1 Motivation

The essence of Peer-to-Peer (P2P) is that nodes in the network directly exploit resources present at other nodes of the network without intervention of any central server. The tremendous success of networks like Napster and Gnutella [1], and of highly visible industry initiatives such as Sun's JXTA [2] , as well as the Peer-to-Peer Working Group including HP, IBM and Intel, have shown that the P2P paradigm is a particularly powerful one when it comes to sharing files over the Internet without any central repository, without centralized administration, and with file delivery dedicated solely to user needs in a robust, scalable manner. At the same time, today's P2P solutions support only limited update, search and retrieval functionality, e.g. search in Napster is restricted to string matches involving just two fields: "artist" and "track". These flaws however make current P2P systems unsuitable for knowledge sharing purposes.

Metadata plays a central role in the effort of providing search techniques that go beyond string matching. Ontology-based metadata facilitates the access to domain knowledge. Furthermore, it enables the construction of semantic queries. Existing approaches of ontology-based information access almost always assume a setting where information providers share an ontology that is used to access the

[1] http://swap.semanticweb.org/

information[3]. In a Peer-to-Peer setting, this assumption does not longer hold. We rather face the situation, where individual peers maintain their own view of the domain in terms of the organization of the local file system and other information sources. Enforcing the use of a global ontology in such a setting would mean to give up the benefits of the Peer-to-Peer approach mentioned above. Therefore, one has to find a way to deal with the existence of multiple, distributed and frequently changing views on the domain.

In this paper, we propose an RDF(S) [4] based metadata model that combines ontological structures with information needed to align, evolve and use these structures for query processing. In section 2, we explain the requirements for the metadata model that guided its development in more detail. Section 3 focuses on the SWAP environment in which the metadata model is used. The model itself is introduced in section 4. In section 5 we describe the methods for applying the metadata model. We conclude with a discussion of open problems.

2 Requirements

We will illustrate the requirements for the proposed system with a short scenario: Virtual organizations or large companies impose a complex situation, with respect to the number of domains, conceptualizations and documents in a peer-system for knowledge sharing. Typically their organizational units are distributed according to expertise or organizational tasks. Subsequently, a case study of a virtual organization in the tourism domain is used as real world example. The virtual organization comprises public authorities, hotels and event organizers. The public authorities require the number of guests visiting the country to plan for example public transport and waste management. Event organizers can customize their offerings according to the number of visitors and their age. Hotels can publish this information to attract more tourists. Today the exchange of this kind of information is time consuming, unpunctual and error prone, although it is often available in electronic form at every level. However, different organizations have diverse objectives and therefore use different conceptualizations of their domains.

From a technical point of view, the different organizations can be seen as one or many independently operating nodes within a "knowledge" network. Nodes can join or disconnect from the network at any moment and can live or act independently of the behavior of other nodes in the system. A node may perform several tasks. Most important is that it acts as a peer in the network, so it can communicate with other nodes to achieve its goals. But apart from that it may act as an interface to interact with the human user of the network, or it may access knowledge sources to accomplish its tasks. One node may have one or more knowledge sources associated with it. These sources contain the information that a peer can make available to other peers. Examples are a user's filesystem and mail folders or a locally installed database.

A node must be designed to meet the following requirements that arise from the task of sharing information from the external sources with other peers:

– Multiple sources of information
– Mostly uniform treatment of internal and external sources
– Multiple views on available information
– Support for query answering and routing
– Distribution of information within the network

The metadata model we will introduce needs to reflect these requirements. We derive objectives for the metadata model with emphasis on information mediation:
Integration: Each piece of knowledge requires metadata about its origin. To retrieve external information, the metadata needs to capture information about where the piece of information was obtained from. This information will allow to identify a peer and locate resources in its repositories.
Information heterogeneity: As each peer uses its own local ontology, the distributed information is inherently heterogeneous. Mappings may be required, e.g. to overcome the heterogenous labelling of the same objects.
Information inconsistency: As information is added from a variety of peers, inconsistencies may occur in a local repository. Information needs to be assigned a confidence rating, such that the system will be able to handle heterogeneity and provide useful information. Similarly, a level of trust can be assigned to peers to model their reliability.
Security: Some information may be of private nature and should not be visible to other peers. Other information may be restricted to a specific set of peers. The metadata model needs to provide means to express these security policies.
Caching: Within Peer-to-Peer systems the availability of other peers is not always guaranteed. Moreover, some peers may have better connectivity, in terms of bandwidth, to the rest of the network than other peers. To improve network efficiency, caching of information can be useful. The caching mechanisms needs to be transparent to the user, but must be captured by the metadata model.

3 The SWAP environment

The SWAP environment is a generic infrastructure which was designed to meet the requirements on a knowledge node. The proposed architecture is shown in figure 1. We will now briefly present the individual components.
Knowledge Sources: Peers may have local sources of information such as the local file system, e-mail directories or bookmark lists. These local information sources represent the peer's body of knowledge as well as its basic vocabulary. These sources of information are the place where a peer can physically store information (documents, web pages) to be shared on the network.
Knowledge Source Integrator: The Knowledge Source Integrator is responsible for the extraction and integration of internal and external knowledge sources into the Local Node Repository. This task comprises (1) means to access local knowledge sources and extract an RDF(S) representation of the stored knowledge, (2) the selection of the RDF statements to be integrated into the Local Node Repository, (3) the annotation of the statements with metadata, and (4)

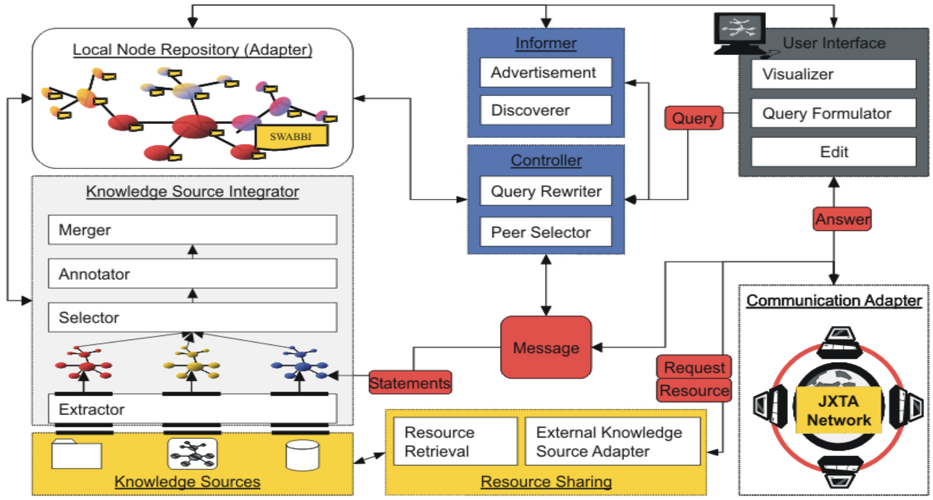

Fig. 1. Abstract Architecture of a SWAP Node

merging the statements into the ontology of the user. These processes, as described in 5.1, utilize the metadata model presented in section 4.1.

Local Node Repository: The Local Node Repository manages an integrated model of the knowledge that is available to the peer locally and remotely. The individual elements of the model are annotated according to the metadata model with their source and a ranking representing the peer's belief in their plausibility.

User Interface: The User Interface of the peer provides individual views on the information available in local sources as well as on information on the network. The views can be implemented using different visualization techniques (topic hierarchies, thematic maps, etc).

Informer: One of the main challenges in a Peer-to-Peer network is that peers initially dont know of each other. The informer provides mechanisms to discover or advertise knowledge pro-actively.

Controller: The Controller is the coordinating component which controls the process of distributing queries. It receives queries from the user interface and distributes them according to the content of the query. When the peer receives a query from another peer, it is tries to answer or forward it. The decision to which peers a query should be sent is made by the Peer Selector based on the knowledge about other peers. Answers received are finally passed back to the graphical interface and to the KnowledgeSource Integrator, which integrates selected content into the Local Node Repository.

Communication Adapter: This component is responsible for the network communication between peers. It serves as a transport layer for other parts of the system, for sending and forwarding queries. It hides and encapsulates all low-level communication details from the rest of the system.

4 The SWAP Metadata Model

The SWAP environment aims at providing a general view on the knowledge each peer has. It facilitates the access to different information sources and enables the user to take advantage from other peers' knowledge. Therefore a metadata model was designed which provides semantics to annotate external as well as internal data. Information from different information sources and from other peers can be integrated with this metadata model to enable a later retrieval of the underlying information items. Furthermore, it allows to cache information to make the entire network work more efficiently. Another purpose of the metadata model is to deal with the information heterogeneity which is inherent in Peer-to-Peer systems.

4.1 Detailed description

As a response to the objectives, we define a SWAP specific metadata model in RDF(S)[5]. An overview of the model is given in figure 2. The complete definition of the model is available at:

`http://swap.semanticweb.org/2003/01/swap-peer#`

The model consists of two RDFS classes, namely the "Swabbi"-class and the "Peer"-class. For these classes, several properties are defined to meet the objectives described above.

Swabbi: Every piece of knowledge can be annoted with a "Swabbi"-object which contains meta-information. It has the following properties:

- *hasPeer:* This property is used to track which peer this "Swabbi"-object is associated with.
- *uri:* Each piece of information was originally created on one peer. To keep track of the origin of the information and to able to unambiguously address an object across the network, the primary URI is explicitly stored with the metadata.
- *location:* Whereas the URI identifies metadata resources within the ontology of the Local Node Repository, the location-attribute is an identifier to access the physical resource, e.g. a document e.g. file://c:/Projects/myfile.txt.
- *label:* The label stores how the specific information is called on the peer it originates from. The label-attribute is expressed in natural language. As one resource can have different names on different peers, this property is added to each "Swabbi"-object and not only to the original object.
- *confidence:* The confidence value indicates - on a scale from 0 to 1 - how reliable a specific statement is. A high confidence in a statement is expressed with high value, 1 meaning the peer is sure the statement is true, 0 meaning the statement is definitely false.
- *security:* Security issues and access rights are important in enterprizes. In the wide open Peer-to-Peer environment some access control is required to ensure proper usage of the information.

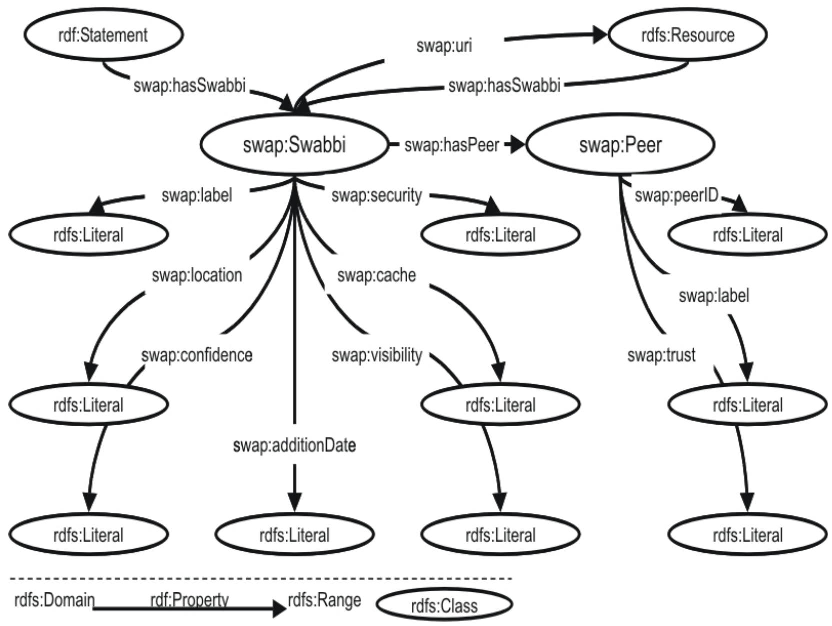

Fig. 2. The SWAP metadata model

- *visibility:* Instead of completely removing objects, they can also be hidden, which can be achieved with this attribute.
- *additionDate:* This attribute keeps track of the date the related resource was added/updated to the Local Node Repository. This could be used to determine confidence; old information might become less reliable.
- *cache:* To increase the network efficiency caching of information will be necessary. The cached information is annotated with this property and set to the date of inclusion.

Peer: For every piece of knowledge we have to remember which peer it originated from. Therefore the Local Node Repository stores different information about each known peer. The information is grouped in the Peer object. The "Swabbi"-object links to the corresponding peer object.

- *peerID:* Each peer has a unique ID to be identified. For our purposes this will be the *JXTA UID*, as we use JXTA as our underlying communication infrastructure,
- *peerLabel:* This peer attribute stores the peer label, which is a human readable and understandable description of the peer in natural language.
- *peerTrust:* Some peers might be more reliable than others. This peer attribute is used to measure trust on a scale from 0 to 1, with 0 meaning having no trust at all and 1 the peer being very trust-worthy.

5 Applying the Metadata Model

In order to use the model described above for semantics-based information exchange, we have to provide a set of methods for constructing the repository according to the metadata model and to assess the knowledge that is stored therein. In the following, we describe methods that have been developed to (1) create and integrate repository content mostly automatically from local and remote information sources, and to (2) rate information in a repository based on the confidence we have in its reliability.

5.1 Knowledge Source Integration

The process of knowledge source integration comprises four subprocesses: This task comprises (1) extraction of an RDF(S) representation from the knowledge sources, (2) the selection of the statements to be included into the Local Node Repository, (3) the annotation of the statements with metadata, and (4) merging the statements into the user's knowledge model.

Extraction As mentioned above, the SWAP-system provides an integrated model of different structures that exists locally on the peer itself and remotely on other peers in the network. These structures are file systems, emails, databases, ontologies and others. In order to include the different information sources, which have been selected by the user for sharing, into our model, we extract an RDF(S) representation from them. The following example shows a fragment of an extracted folder structure – a resource of type Folder with the label "Project" and the original location in the folder structure:

```
<rdf:RDF xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#'
    xmlns:rdfs='http://www.w3.org/2000/01/rdf-schema#'
    xmlns:swapcommon="http://swap.semanticweb.org/2003/01/swap-common#"
    xmlns:swap="http://swap.semanticweb.org/2003/01/swap-peer#">
<swapcommon:Folder
        rdf:about="swap://1234567890.jxta#project">
    <rdfs:label xml:lang="en">Project</rdfs:label>
    <swapcommon:location>
        filefolder://windows/c:/Project
    </swapcommon:location>
</swapcommon:Folder>
</rdf:RDF>
```

In a second step, this extracted information can be semantically enriched. This means adding formerly implicit semantics explicitly to the structures as theoretically described by [6] as emergent semantics. We assume for the moment that we can determine the meaning, as intended by the user, and type of a certain information item in an ontological sense and can describe it as either concept, instance or property.

If we extract a "Project"-folder from our file-system and this folder has a subfolder "SWAP" we assume that the user placed these items this way, because SWAP is an instance of project. This aspect is modelled as follows:

```
<rdf:Description rdf:about="swap://1234567890.jxta#SWAP">
    <rdf:type rdf:resource="swap://1234567890.jxta#project"/>
</rdf:Description>
```

Selection The selection process can be described as a filter on incoming statements. It has to decide which information is useful and therefore should be included into the Local Node Repository. Metadata, e.g. previously collected metadata about the trust in other peers, might prove useful in defining the selection process. The information selected to be included in the Local Node Repository will be annotated with metadata, as described in the next paragraph.

Annotation Having built an RDF(S) representation of the information sources, in this processing step the "Swabbi"-objects are added. We have to distinguish between information that originated from the extraction process on the local peer and information received from other peers. For extracted structures, a new "Swabbi"-object is created for each resource or statement. To link a "Swabbi"-object to a statement we use the RDF construct of reification. The properties (as presented in 4.1) are filled accordingly and a reference between the resource and the "Swabbi"-object is established with a "hasSwabbi" relation.

The other major source of information are other peers. The selection process has already selected the information to keep. This information is added to the Local Node Repository in the same way as any other information from the peer itself. Additionally, information about the originating peer is included using the 'hasPeer" property of the "Swabbi"-object. The example shows a reified statement with its "Swabbi" information – the first element describes the statement itself, the second one describes the associated "Swabbi"-object, and the last one describes the corresponding peer.

```
<rdf:Statement rdf:about="swap://1234567890.jxta#statement01">
    <rdf:subject rdf:resource="swap://1234567890.jxta#project"/>
    <rdf:predicate rdf:resource="rdfs:subclassOf"/>
    <rdf:object rdf:resource="swap://1234567890.jxta#thing"/>
    <swap:hasSwabbi rdf:resource="swap://1234567890.jxta#swabbiObjectNo01"/>
</rdf:Statement>
<swap:Swabbi rdf:about="swap://1234567890.jxta#swabbiObjectNo01">
    <swap:hasPeer  rdf:resource="swap://1234567890.jxta#knownPeers0001" />
    <swap:label>Project</swap:label>
    <swap:uri rdf:resource="swap://1234567890.jxta#project" />
    <swap:location>filefolder://windows/c:/Project</swap:location>
</swap:Swabbi>
<swap:Peer rdf:about="swap://1234567890.jxta#knownPeers0001">
    <swap:peerId>1234567890</swap:peerId>
    <swap:peerLabel>Christoph</swap:peerLabel>
</swap:Peer>
```

Merging One goal of the SWAP system is to have a *single* model. Through addition of resources and statements from other peers the same object might be present in the local repository, but under different names.

The system has to identify these two objects through similarity measures. In the approach of [7] three methods are presented, which we extend with a fourth one:

- Word Matching: By checking the labels one can determine if the objects are similar (e.g. by calculating the edit distance [8]). Alternatively WordNet can be used to find synonyms.
- Feature Matching: If an entity has the same predicates and objects as another one this is a strong indication that the object is the same in both cases.
- Semantic-Neighborhood Matching: This method checks the (contextual) neighborhood of two entities ([9]).
- Instance Matching: We also look for instances and where they are classified in the structure. This indicates similarity of the respective classes.

The first prototype of the SWAP system uses word matching comparisons based on the labels of entities. Instance matching is realized by comparing unique identifiers, e.g. a hash code in the case of files and email addresses in the case of persons. Ongoing efforts are being made to also include the other methods, keeping in mind that some have to be adapted to keep the computational complexity manageable.

The last step is to actually do the merging in the Local Node Repository. Whereas in RDF(S) no explicit relation as "equals" is defined, we derived a relation from the equality relation defined in OWL [10].

5.2 Content Rating Model

Statements made by peers can be incomplete, vague or even false. For this reason, statements are not accepted by a peer as an absolute truth, but are judged based on the previous experience with the sender [11]. For example, if the sender tells the receiver something about youth hostels and the receiver knows that the sender is an expert on hotels, then it can derive from the fact that both concepts (hotel and youth hostel) have a small semantic distance that the sender probably also knows more than an average user about youth hostels. To formalize the expertise we introduce confidence ratings that are meta-statements placed in the 'Swabbi'-object that indicate the confidence in a certain statement. We will now describe the different aspects of the rating methods:

Assigning confidence ratings to statements from a peer Here we have to distinguish between derived statements from the extraction algorithm described in the previous paragraph and statements received from external peers. In the first case we assume that the user is confident in the statements that are derived and therefore assigned a high confidence rating. In the second case the confidence ratings are calculated from the previous statements from the sender. When a peer a receives information from peer b and b is unknown to a, then the statement from b gets a (low) initial confidence rating. If, however b already provided statements before to a then the new confidence is calculated out of these statements. The value is a weighted average where the weighting factor is determined by semantic distance between an old statement and the new one. The similarity measure we use is adapted from [12].

Updating confidence ratings If other peers that are different from the original sender confirm the statement by repeating it, the statement gains higher confidence. The amount of gain depends of the confidence in the confirming source. This recursive definition of rating is also used in PageRank in Google [13] where the rank of a source depends on the ranks of the sources voting for that source.

Determining the experts to be queried When a query is received, the receiver first tries to answer the query itself. If it doesn't have a satisfying answer, it tries to find experts on the topic of the query based on the information that has been received from other peers. The system tries to find experts on topics that have a close semantic distance to the topic of the query. Again, we use the similarity measure described above for this aspect.

Aging mechanism to devaluate confidence ratings in time A SWAP peer can retrieve large set of statements from other peers and from the generated statements by the ontology extractor. To keep the local repository scalable, we use an aging mechanism that removes statements that are too old in combination with a low rating.

6 Related Work

Knowledge management in Peer-to-Peer systems is the topic of various active research projects. Edutella [14], [15] provides an RDF-based infrastructure for exchanging metadata in P2P applications. The Edutella Query Service is intended to be a standardized query exchange mechanism for RDF metadata stored in distributed RDF repositories. The Edutella project focuses on the education community. The Edamok project [16] also deals with distributed knowledge management in Peer-to-Peer systems. It does not use an ontology premise though.

Emergent Semantics [17] builds on lightweight (e.g. a file structure with files as instances) and/or heavyweight ontologies that different participants have created. [18] describes an approach for obtaining semantic interoperability among data sources without relying on pre-existing, global semantic models. It enables the participating data sources to incrementally develop global agreement in a completely decentralized process that relies on pair-wise, local interactions, such as in P2P-systems.

7 Conclusion

The completely distributed nature and the high degree of autonomy of individual peers in a P2P system comes with new challenges for the use of semantic descriptions. If we want to benefit from the advantages that normally accompany the use of ontologies as specifications of a shared vocabulary we have to find ways to dynamically align the semantic models of different peers. In this paper, we described a model that combines features of ontologies with rich metadata about the origin of information and the reliability of sources. Furthermore, we introduced methods for creating, integrating and assessing metadata.

Our model has several advantages over traditional ontologies in the context of Peer-to-Peer information exchange. The most important feature is the fact that statements in the semantic models are not seen as being the truth as in most traditional models. We rather see the semantic model as a collection of opinions supported by different sources of information. Opinions that many sources agree on are more likely to be true than opinions that are not shared across the system or that even contradict with other opinions. This makes it possible to directly extract semantic models from information sources even if these are not completely compliant with the existing model. Furthermore, we can use heuristic methods to align and update semantic models. If the result of such an update is shared by many peers it will persist.

A key issue for the acceptance of such heuristic methods of course is a careful evaluation of their performance in general and in concrete applications. In order to evaluate the model and the methods on a general level, test procedures are developed in the SWAP project that use simulation techniques to experiment with large scale Peer-to-Peer systems [19]. Furthermore, case studies for the SWAP system in the tourism and finance domains are planned. These case studies will show how much benefit the developed methods provide in real world applications.

One of the most fundamental questions that has to be answered by the case studies is whether it is sufficient to rely on structures that have been extracted from information sources instead of hand-crafted knowledge structures and metadata. It may turn out that in addition to the extraction approach, we also need to annotate information by hand. In this case we have to investigate how methods for supporting semantic annotation can be integrated in the system in order to build up the knowledge structures described in this paper.

8 Acknowledgements

SWAP is a project funded by the EU under contract no. EU IST-2001-34103.

References

1. Kan, G.: Gnutella. In: Peer-to-Peer: Harnessing the Power of Disruptive Technologies. O'Reilly (1999) 94–122
2. Traversat, B., Abdelaziz, M., Duigou, M., Hugly, J.C., Poulouy, E., Yeager, B.: Project JXTA Virtual Network. Technical report, Sun Microsystems Inc. (2002)
3. Gruber, T.R.: Towards Principles for the Design of Ontologies Used for Knowledge Sharing. In Guarino, N., Poli, R., eds.: Formal Ontology in Conceptual Analysis and Knowledge Representation, Deventer, The Netherlands, Kluwer Academic Publishers (1993)

4. Brickley, D., Guha, R.: RDF Vocabulary Description Language 1.0: RDF Schema (1999)
http://www.w3.org/TR/rdf-schema/.
5. Broekstra, J., Klein, M., Decker, S., Fensel, D., van Harmelen, F., Horrocks, I.: Enabling knowledge representation on the web by extending rdf schema. In: Proceedings of the tenth World Wide Web conference WWW'10, Hong Kong (2001)
6. Maedche, A., Staab, S.: Ontology learning for the semantic web. IEEE Intelligent Systems **16** (2001) 72–79
7. Rodrguez, M.A., Egenhofer, M.J.: Determining semantic similarity among entity classes from different ontologies. IEEE Transactions on Knowledge and Data Engineering **15** (2003) 442–465
8. Levenshtein, I.V.: Binary codes capable of correcting deletions, insertions, and reversals. Cybernetics and Control Theory (1966)
9. Maedche, A., Motik, B., Silva, N., Volz, R.: Mafra - a mapping framework for distributed ontologies. In: Proceedings of the EKAW (Knowledge Engineering and Knowledge Management) 2002. Volume 2473 of Lecture Notes in Computer Science., Springer (2002)
10. Dean, M., Connolly, D., van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D.L., Patel-Schneider, P.F., Stein, L.A.: Owl web ontology language 1.0 reference, Internet:http://www.w3.org/TR/owl-ref/. (2002)
11. Siebes, R., van Harmelen, F.: Ranking agent statements for building evolving ontologies. Proceedings of the AAAI-02 workshop on meaning negotiation, Alberta, Canada (2002)
12. Resnik, P.: Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. Journal of Artificial Intelligence Research **11** (1999) 95–130
13. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project (1998)
14. Nejdl, W., Wolf, B., Qu, C., Decker, S., Sintek, M., Naeve, A., Nilsson, M., Palmér, M., Risch, T.: Edutella: A P2P networking infrastructure based on rdf. In: Proceedings to the Eleventh International World Wide Web Conference, Honolulu, Hawaii, USA (2002)
15. Nejdl, W., Wolpers, M., Siberski, W., Schmitz, C., Schlosser, M., Brunkhorst, I., Lser, A.: Super-peer-based routing and clustering strategies for rdf-based peer-to-peer networks. In: Proceedings of the 12th International World Wide Web Conference, Budapest, Hungary. (2003)
16. Bonifacio, M., Cuel, R., Mameli, G., Nori, M.: A peer-to-peer architecture for distributed knowledge management. In: Proceedings of the 3rd International Symposium on Multi-Agent Systems, Large Complex Systems, and E-Businesses MALCEB'2002, Erfurt, Germany. (2002)
17. Maedche, A.: Emergent semantics for ontologies - support by an explicit lexical layer and ontology learning. IEEE Intelligent Systems - Trends and Controversies **17** (2002) 78–86
18. Aberer, K., Cudré-Mauroux, P., Hauswirth, M.: The Chatty Web: Emergent Semantics Through Gossiping. In: Proceedings of the 12th International World Wide Web Conference, Budapest, Hungary. (2003)
19. Ehrig, M., Schmitz, C., Staab, S., Tane, J., Tempich, C.: Towards evaluation of peer-to-peer-based distributed knowledge management systems. In: Proceedings of the AAAI Spring Symposium "Agent-Mediated Knowledge Management (AMKM-2003)". (2003)