AN ONTOLOGY-BASED APPROACH FOR SEMANTIC INTEROPERABILITY IN P2P SYSTEMS

Deise de Brum Saccol, Rodrigo Perozzo Noll, Nina Edelweiss and Renata de Matos Galante
Instituto de Informática, Universidade Federal do Rio Grande do Sul,
Bento Gonçalves 9500, Porto Alegre, Brazil

Keywords: Ontology, peer-to-peer, schema matching, similarity.

Abstract: In peer-to-peer (P2P) systems, files from the same application domain are spread over the network. When the user poses a query, the processing relies mainly on the flooding technique, which is quite inefficient for optimization purposes. To solve this issue, our work proposes to cluster documents from the same application domain into super peers. Thus, files related to the same universe of discourse are grouped and the query processing is restricted to a subset of the network. The clustering task involves: ontology generation, document and ontology matching, and metadata management. In this paper, we focus on the matching step.

1 INTRODUCTION

P2P refers to a class of application systems that use distributed resources to perform tasks in a decentralized context[1,2,3]. The usability of such systems is mainly dependent on techniques used to find and retrieve results. However, the searching optimization faces two problems: how to find relevant files for the user query with a low cost, and how to deal with the poor semantics of the resources.

Files that belong to the same application domain and that are needed for answering a specific query may be stored in several peers. The use of the flooding technique would be necessary in order to access all the relevant files. But this technique is expensive and time-consuming, since all the peers get the query message and usually only some of them are able to answer it. Another attempt for balancing cost is to use any flooding variation technique, such as *breadth-first* traversal (over the network with depth limit *D* measured in hops) or *depth-first* traversal with depth limit *D* (Yang, 2002). However, their use does not guarantee the optimal results, since not all the peers get the requesting message.

Related to the semantics, let us consider two applications that need to exchange data. One possible approach is to build an adapter that transforms data and structure between them. However, the adapter construction is a hard task that requires knowing the data organization in both applications. Furthermore, the complexity and the developing time tend to be quadratic in relation to the number of component applications (Staab, 2004). A possible solution should use some kind of metadata for describing the semantics of the underlying repositories. But this scenario states two critical questions (Mena, 2001): (i) how to deal with different concepts used to describe the same information; and (ii), how to acquire and maintain the necessary metadata to solve the vocabulary sharing issue.

To overcome these problems we propose an ontology-based approach that can be used for improving traditional searching techniques in P2P systems. We focus on two important issues: (i) the extra traffic produced by traditional flooding techniques when the optimal results are required; and (ii) the lack of semantics regarding the information storage and searching. In order to avoid such unnecessary traffic caused by the flooding solution, our approach relies on file clustering into super peers, based on the application domain of the peer files. The file grouping criteria is based on the

[1] Gnutella. Available at http://www.gnutella.com
[2] Kazza. Available at http://www.kazaa.com
[3] eMule. Available at http://www.emule-project.net

de Brum Saccol D., Perozzo Noll R., Edelweiss N. and de Matos Galante R. (2008).
AN ONTOLOGY-BASED APPROACH FOR SEMANTIC INTEROPERABILITY IN P2P SYSTEMS.
In *Proceedings of the Tenth International Conference on Enterprise Information Systems - DISI*, pages 310-316
DOI: 10.5220/0001711503100316

similarity analysis between the files and the domain. In order to increase the semantics, we adopt an ontology for describing the document concepts.

The main contributions of this paper are:

- The specification of the ontology manager, a component used for allowing semantic interoperability in P2P systems;
- The definition/implementation of a mechanism for file and ontology matching, based on lexical and semantic similarity between concepts.

The functionalities of this mechanism are performed by the ontology manager, as part of *DetVX* (Saccol, 2007), as following described.

2 *DeTVX* FRAMEWORK

The proposed framework (***Det**ector of Replicas and **V**ersions of **X**ML Documents*) stores files according to the super peer architecture. Files stored in peers are related to a specific application domain, described by an ontology (e.g., *curriculum* or *research projects* domain). We use the ontology as peer grouping criterion into super peers. The ontologies are automatically generated (from the schema integration process). All the peers belonging to the same domain are clustered in the same super peer. Since one domain is represented in only one super peer, two files belonging to the same domain cannot be found in different super peers. With this assumption, a certain query related to a domain will be forwarded only within a specific super peer network, reducing unnecessary network traffic.

2.1 Ontology Manager

The ontology manager is responsible for maintaining the ontology repository and for associating ontologies to super peers. In this proposal, ontologies are represented in OWL[4] format. The activities are presented in Figure 1 and are described as follows:

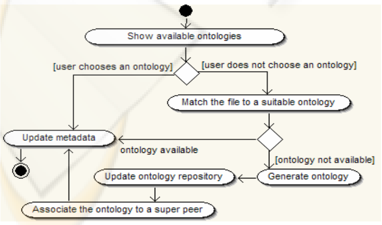
Show available ontologies
[user chooses an ontology]
[user does not choose an ontology]
Match the file to a suitable ontology
Update metadata
ontology available
[ontology not available]
Update ontology repository
Generate ontology
Associate the ontology to a super peer

Figure 1: UML activity diagram for the ontology manager.

- *Show available ontologies* – this activity returns a short description of existent ontologies stored in the repository. The user can choose an ontology to be used for a (set of) document (s) belonging to this domain (i.e., select a super peer to connect).
- *Match the file to a proper ontology* – for document and ontology matching, our work assumes one of the following approaches:
  - There is a compatible ontology for the document – thus, the peer that stores the file is connected to the super peer described by this ontology. To figure out which ontology best describes a certain file, our proposal relies on using similarity techniques between the file and the ontologies. The ontology that presents the higher similarity is chosen to represent the file domain, as described in Section 3.
  - There is not a compatible ontology for the document – thus, we create a new ontology (*generate ontology* activity) that represents the concepts and relationships of this domain.
- *Generate* ontology - The ontology is created from the schema integration process. The integration activity uses a *thesaurus* for helping to infer terminological relationships in order to match equivalent concepts. The full description of this task is presented in (Saccol, 2008).
- *Update ontology repository* – this activity is responsible for storing the created ontology into the ontology repository.
- *Associate the ontology to a super peer* – this activity is responsible for associating the new ontology to an existent super peer. Some administrative metadata also need to be maintained.

Load balancing must be considered when choosing a super peer to associate to an ontology. In our work, load balancing is performed based on the number of peers in super peers. The goal is to keep the same number of peers in each super peer. The mean number (A) of peers (p) in each super peer is defined as: *A(p)=(number of peers)/(number of super peers)+range*. The range value allows some flexibility in terms of number of peers before performing load balancing. The load balancing must agree with the grouping criteria (belong to the same ontology). Thus, the load balancing regrouping may imply on associating ontologies to other super peers.

- *Update metadata* – this activity updates the metadata.

In this paper, we focus on file and ontology matching.

[4] Web Ontology Language (OWL). Available at http://www.w3.org/2004/OWL/

3 ONTOLOGY AND DOCUMENT MATCHING

The matching task aims to define the ontology that best describes a XML file, by measuring the overall similarity between both representations. Given a XML file *d* and a set of *n* ontologies $O=\{o_1,o_2,o_3,...,o_n\}$, the procedure computes the similarity score *sim(d,O)*. The ontology o_m (0<m<=n) with the highest score (greater than a threshold *t)* is chosen to represent the corresponding file domain application.

The strategy aims to find resemblances between classes. Two questions must be answered: 'which pairs of classes will be compared' and 'what are the criteria to determine how similar the classes are'.

Several approaches address similarity analysis (Madhavan, 2001), (Maedche, 2002). These approaches are usually based on three main steps:

- Normalization: determines which elements are semantically equivalent;
- Categorization: separates the elements into classes, in order to reduce the number of comparisons;
- Comparison: defines the similarity score computed among the elements in their categories.

To evaluate the similarity between files, two types of perspectives are considered: the lexical perspective evaluates the relations between terms by comparing the strings, while the semantic perspective focuses on the meaning and conceptual correlation among terms. For the similarity analysis, we consider both types, as follow described:

- **Lexical Similarity Analysis**: two main approaches are used:
  - ○ Edit distance functions (Levenshtein, 1966): this approach analyses the minimum number of operations to transform one character sequence into another;
  - ○ Stemmer algorithms[5]: this approach reduces the character sequence to the stem (i.e., the form of a word after all affixes are removed).

  In our approach, we use a *stemmer* algorithm for lexical similarity analysis.
- **Semantic Similarity Analysis**: two main approaches are commonly used:
  - ○ *Thesaurus*: used to figure out the terminological relationships (e.g., *WordNet*[6]).
  - ○ Taxonomic Overlap procedure (Maedche, 2002): it does not individually analyze the element, but the element context. For calculating the similarity degree between two sets of elements, we use the *Jaccard* coefficient (Manning, 1999).

Our mechanism aggregates and extends the advantages of some existent approaches, as described in the next section.

3.1 Matching Approach

We use the following process to compute the similarity score $sim(d,o_n)$ between a XML file *d* and an ontology o_n. The first step corresponds to the normalization and categorization tasks, as follows.

- Normalization and Categorization Phase: for each file (XML and OWL), we map all the component elements, by traversing the document and storing the stem (key) and a list with the complete names for each element. If the element name is composed by *n*-words, then the list also consists of each component. This initial mapping corresponds to the lexical perspective and is presented in Figure 2.

Key	List		
skill	skillArea	skill	area

Figure 2: Lexical normalization of XML elements.

The next step considers the element synonyms. The list of synonyms is retrieved from *WordNet*. The resulting list is presented in Figure 3.

Key	List					
skill	skillArea	skill	area	accomplishment	acquisition	domain

Figure 3: Semantic normalization of XML elements.

The normalization occurs in two steps. First, we normalize the ontology elements, as above described. We traverse the XML elements and verify in their lists the existence of any lexical correspondence with other elements from the OWL list. These correspondences are analyzed by looking only at the stems, using a *stemmer*. At last, we have two normalized and categorized lists (XML and OWL elements). In this phase, the system is able to identify the existent correspondences between the XML and OWL elements, providing the initial step for element comparison.

- Comparison Phase: this step analyzes the taxonomic overlap. For each element that consists of a *root-leaf* set (i.e., a path from the root to the leaf

[5] The Lancaster Stemming Algorithm. Available at http://www.comp.lancs.ac.uk/computing/research/stemming

[6] WordNet. Available at http://wordnet.princeton.edu

nodes), the system knows the existence of correspondences in the ontology (obtained during the normalization task). To get the similarity degree, we define the union as the total number of XML elements in the *root-leaf* set. The intersection is defined as the number of relations with a correspondence between the XML and the ontology elements. By relations, we mean: an OWL class that relates to an OWL subclass; an OWL class that relates to *Object* or *Datatype* properties; and the relations between OWL classes.

The taxonomic overlap method traverses the left-side tree and analyzes only the nodes that have correspondence with the right-side tree. This procedure is repeated for all the *root-leaf* sets. Thus, we obtain a list with the all the similarity values. The final value is calculated as the arithmetical mean of this list.

3.2 Matching Example

Let us consider the XML file presented in Figure 4.

```
<resume><header>
        <name><firstname>Jo</firstname>
        <surname>Doe</surname></name>
        <address><street>123 Elm #456</street>
        <city>Garbonzoville</city><state>NX</state>
        <zip>99999-9999</zip></address>
        <contact><phone>555.555.5555</phone>
        <email>doe@doe.doe</email>
<url>http://doe.com/~doe/</url></contact>
</header></resume>
```

Figure 4: XML document.

Figure 5 presents part of the *resume* ontology (some properties are not shown, such as the address properties - city, state, street and zip). Let us also suppose the existence of another ontology representing another domain, e.g., academic research. We aim to figure out which ontology best describes the XML document, by measuring the similarity among the XML file and the ontologies.

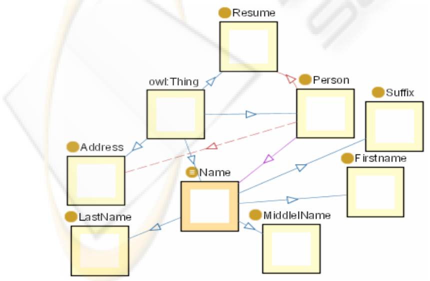

Figure 5: The *resume* ontology.

By following the mechanism previously presented in Section 3.1, the matching approach produces the similarity values described in Table 1. By looking at the last line, we can see that the first ontology (*resume.owl*) presents the highest similarity (0.899). Thus, this ontology is chosen for describing the XML file domain.

While analyzing the partial similarities, the mechanism is capable to store information about mappings between concepts in the XML document and concepts in the ontology. Basically, for each term in the ontology the mapping information is represented as a transforming function. The transforming functions are represented in *XPath*[7]. For instance, the concept *fullName* can be mapped in different documents, such as *name* and *firstName+lastName*. The mappings are extensively used in query transformation.

Table 1: Individual (ISim), partial and final similarity values for the files presented in Figure 4 and Figure 5.

XML (Resume)	Ontology (Resume)	ISim	Ontology (Academic Research)	ISim
[header, name, firstname]	[Name, Firstname]	0.667	[firstName, firstName]	0.333
[header, name, surname]	[Name, LastName]	0.667	[firstName, lastName]	0.333
[header, name]	[name]	0.500	[name]	0.500
[header, address, street]	[Address, street]	0.667	[address]	0.333
[header, address, city]	[Address, city]	0.667	[address]	0.333
[header, address, state]	[Address, state]	0.667	[address, Department]	0.667
[header, address, zip]	[Address, zip]	0.667	[address, nil]	0.333
[header, address]	[address]	0.500	[address]	0.500
[header, contact, phone]	[phone]	0.333	[phone]	0.333
[header, contact, email]	[email]	0.333	[email]	0.333
[header, contact, url]	[url]	0.333	[url]	0.333
[header, contact]	[]	0.000	[]	0.000
[header]	[]	0.000	[]	0.000
…	…	…	…	…
Partial similarity value		**0.462**		**0.333**
Final similarity value		**0.899**		**0.433**

Some metadata are also maintained, but not presented in this paper.

4 IMPLEMENTATION

Our approach is implemented in a tool named *The Matcher*. The tool measures the lexical and semantic similarity among XML files and ontologies. *The Matcher* was implemented using the following APIs: *Paice/Husk* stemmer [8], for the lexical analysis (word

[7] XML Path Language. Available at http://www.w3.org/TR/xpath20/

[8] The Lancaster Stemming Algorithm. Available at http://www.comp.lancs.ac.uk/computing/research/stemming

stems); *WordNet*[9], for the semantic analysis (synonyms); JWNL[10], for accessing the *WordNet* dictionary; *Jena*[11], for manipulating the OWL files; and *Xerces*[12], for manipulating the XML files.

We ran some experiments to calculate the similarity between a curriculum XML file and some ontologies. The goal is to prove that a file belonging to a domain *D* has higher similarity with the ontology that describes that domain than others. The input consists of one XML file and six OWL files. The output is a similarity value for each pair of representations. The ontologies are related to the following domains: academic research, amino acids, wine, pets and owners, travel, and curriculum. The similarity results are presented as follows.

Academic_research.owl: 0.496	Amino_acid.owl: 0.377
People_pets.owl: 0.448	Resume.owl: 0.899
Travel.owl: 0.382	Wine.owl: 0.39

The similarity values show that the ontology Resume.owl has the highest similarity with the XML document used in the experiments (89.9%). Other ontologies from different domains presented very low similarities (between 38% and 49%). Thus, the *Resume* ontology is chosen for describing the document. In these experiments, we considered that ontology with the highest similarity is always chosen. However, it may occur that even the most similar ontology is still not proper to describe a document (i.e., low similarity). In this case, our proposal relies on a threshold-based approach. The threshold definition is not addressed in this paper. Our mechanism allows users to formulate queries over ontologies and the system takes the responsibility for managing the heterogeneity and distribution in the peers.

5 RELATED WORK

The usability of P2P systems is mainly dependent on the techniques used for finding and retrieving the resources. The result quality can be measured by some metrics (Yang, 2002), such as: the size of the result set, the user satisfaction, and the time to get the results. However, there is a relation between cost and quality that must be balanced. The use of the flooding technique guarantees the optimal results, but it is expensive and time-consuming, since all the peers get the query message and usually only some of them are able to answer it. Another attempt for balancing cost is to use any flooding variation techniques, such as *breadth-first* traversal[13] (over the network with depth limit *D* measured in hops) or *depth-first* traversal[14] with depth limit *D*. However, their use does not guarantee the optimal results, since not all the peers get the requesting message.

For systems focusing on availability, search techniques such as those presented in (Ratnasamy, 2001) and (Rowstron, 2001) are well-suited, because they guarantee location of content if it exists. However, to achieve these goals, these techniques strongly control the data placement and work only for a delimited number of hops. In our proposal, it is fundamental to retrieve all the relevant files, while using an optimized searching technique.

Furthermore, keyword-based systems do not retrieve the necessary document if a synonym is used as part of the query. This situation happens because different but related terms may be used to describe similar information. Besides, automatic systems lack to find and extract relevant information and fail to integrate information spread over different sources (Fensel, 2001). To address these issues, semantic annotations in our proposal allow structural and semantic definitions of documents, providing an intelligent query processing that allows users to pose queries in a P2P system without being aware of the location and structure of the files.

The peer aggregation into super peers is an important issue and it is the basis of our proposal for searching enhancement. This task is usually performed based on some features, such as subject and location (Nejdl, 2002). In this paper, we consider both the domain application of the files (described by an ontology) and a quantitative metric defined by the desired mean number of peers in super peers.

There are some works based on similarity techniques that can be used for figuring out the application domain of a XML document (Bertino, 2004). These works are usually based on structural (Francesca, 2003), (Dalamagas, 2004), (Lian, 2204) or content similarity (Baeza-Yates, 1999). *Diff* algorithms are commonly used for detecting differences between the files. However, our problem is to identify semantic correspondences, which may exist even between representations that are quite differently in structure and content. Although an ontology and a XML file can present low structural and content similarity, they can describe the same application domain. Thus, traditional similarity techniques are not proper for our problem.

[9] WordNet. Available at http://wordnet.princeton.edu
[10] JWNL Java WordNet Library. Available at http://jwordnet.sourceforge.net
[11] Jena-Semantic Web Framework. Available at http://jena.sourceforge.net
[12] Xerces. Available at http://xerces.apache.org

[13] Gnutella. Available at http://www.gnutella.com
[14] Freenet. Available at http://freenet.sourceforge.net

6 CONCLUDING REMARKS

Search engines provide support for automatic information retrieval which helps in finding data sources. However, the tasks of extracting the relevant information remain for the user. Thus, there are some bottlenecks that must be passed, such as (Fensel, 2001): lack of a means for representation and translation and lack of a means for content descriptions.

Considering P2P systems, there is an extra issue: to increase the result quality while optimizing the search space. In this scenario, two problems must be addressed: how to find relevant files for the user query and how to increase the semantics in the information resources. To overcome these issues, we proposed an ontology-based approach that can be used for improving searching techniques. With this proposal, we have reduced the extra traffic produced by traditional flooding techniques when the optimal results are required, and increased the semantics regarding the information storage and searching. The search space optimization is achieved by clustering files into super peers, based on file similarity. The increasing of the semantics is done by adopting ontologies, making explicit the information content in a manner independent of the underlying structures used to store the information.

We have presented the ontology manager, by defining and implementing a tool for matching ontologies to XML documents. By matching the ontology to a XML file, the system can connect the peer to a proper super peer that is described by a specific ontology. The matching phase basically considers the concept name, the structure similarity and stemmer algorithms. The ontologies are generated from an integration process among the conceptual schemas that describe the XML files.

We implemented a tool named *The Matcher* that identifies the similarity between XML files and OWL ontologies. To evaluate the results, we have performed a set of experimental tests, which clearly demonstrated the accurate results. As future work, we are going to incorporate this tool into *DetVX*, a framework for detecting, managing and querying XML replicas and versions in P2P scenarios. We are currently developing a graphic tool for peer management based on JXTA platform (Gong, 2001). The system will allow managing the super peers, peers and corresponding files, as well to assess the performance when using the presented approach.

ACKNOWLEDGEMENTS

This work has been partially supported by CNPq under grant No. 142396/2004-4 for Deise de Brum Saccol; Pronex Project – FAPERGS under grant No. 0408933 and CNPq Universal under grant No. 481055/2007-0 for Renata de Matos Galante.

REFERENCES

Baeza-Yates, R.A. and Ribeiro-Neto, B.A., 1999. Modern Information Retrieval. ACM Press / Addison-Wesley.

Bertino, E.; Guerrini, G. and Mesiti, M., 2004. A Matching Algorithm for Measuring the Structural Similarity between an XML Document and a DTD and its Applications. Information Systems, Elsevier Science Ltd., 29, 23-46.

Dalamagas, T.; Cheng, T.; Winkel, K.J. and Sellis, T.,2004. Clustering XML Documents using Structural Summaries. In: EDBT Work. on Clustering Information over the Web, Greece.

Fensel, D., 2001. Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce. Springer.

Francesca, F.D.; Gordano, G.; Ortale, R. and Tagarelli, A.., 2003. Distance-based Clustering of XML Documents. In: Work. on Mining Graphs, Trees and Sequences, Croatia.

Gong, L., 2001. JXTA: A Network Programming Environment. IEEE Internet Computing, 5(3):88–95, May/June.

Kantrowitz, M., Mohit, B. and Mittal, V., 2000. Stemming and its effects on TFIDF ranking. In: SIGIR Conf. on Research and Development in Information Retrieval. Athens.

Levenshtein, V., 1966. Binary Codes capable of correcting deletions, insertions, and reversals. Cybernetics and Control Theory, 10(8):707–710.

Lian, W.; Cheung, D.; Mamoulis, N. and Yiu, S., 2004. An Efficient and Scalable Algorithm for Clustering XML Documents by Structure. IEEE Trans. on Knowledge and Data Engineering , 16, 82-96.

Madhavan, J., Bernstein, P. A. and Rahm, E., 2001. "Generic schema matching using Cupid". In: VLDB'01, Rome, Italy.

Maedche, A.; Staab, S., 2002. "Measuring similarity between ontologies". In: EKAW.

Manning, C. D. and Schütze, H., 1999. Foundations of Statistical Natural Language Processing. 1st ed. Cambridge, MA: MIT Press.

Mena, E., and Illarramendi, A., 2001. Ontology-based query processing for global information systems. Springer.

Nejdl, W., Wolf, B., Qu, C., Decker, S., Sintek, and et. al., 2002. EDUTELLA: A P2P Networking Infrastructure Based on RDF. In: WWW'02, Hawaii, EUA.

Ratnasamy, S., Francis, P., Handley, M., Karp, R., and Shenker, S., 2001. A scalable content-addressable network. In: SIGCOMM.

Rowstron, A., and Druschel, P., 2001. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In: Middleware.

Saccol, D.B., Edelweiss, N. , Galante, R.M. and Zaniolo, C., 2007. Managing XML Versions and Replicas in a P2P Context. In: SEKE, Boston, USA.

Saccol, D.B. et al., 2008. Gerenciamento de Domínios de Aplicação através do Uso de Ontologias. In: ERBD, Florianópolis, Brazil – in Portuguese. (to be presented).

Staab. S. and Studer, R., 2004. Handbook on Ontologies (International Handbooks on Information Systems). Springer.

Yang, B. and Garcia-Molina, H., 2002. Efficient Search in Peer-to-Peer Networks. In: ICDCS, Vienna, Austria.