SemRef: A Semantic-based Query Reformulation Tool for Dynamic Environments

Damires Souza[1,2], Thiago Arruda[1], Ana Carolina Salgado[1], Patricia Tedesco[1]

[1]Centro de Informática/UFPE, Brazil

[2]Instituto Federal de Educação, Ciência e Tecnologia da Paraíba/IFPB, Brazil

{dysf,tan, acs, pcart}@cin.ufpe.br

Abstract. One key issue for query answering in dynamic environments is the reformulation of a query posed at a peer into another one over a target peer. In this paper, we present a query reformulation tool - named *SemRef*, which uses semantics to enhance query reformulation through query enrichment and provides users with a set of expanded answers. The semantics is acquired from a set of mappings (that extend the ones commonly used) and from the context of the user, of the query and of the environment. The tool's interface enables users to submit queries using patterns from both ALC/DL and SPARQL query options. The tool also provides logs which show how query reformulation is accomplished and query answers are produced. Thus, administrators can verify the correctness of both tasks, while users benefit from query enrichment.

1. Introduction

Query answering has been addressed as a key issue in dynamic environments such as Peer Data Management Systems (PDMS) [Adjiman et al. 2007]. An important step in this process is reformulating a query posed at a peer (data source) into a new query expressed in terms of a target peer, considering existing mappings (here called *correspondences*) between them. In this light, query reformulation approaches have received a great deal of attention from the database community research (e.g., [Kostadinov 2007], [Stuckenschmidt et al. 2005] and [Necib and Freytag 2005]). However, a problem which remains unanswered is how to exploit these correspondences in order to improve query reformulation in such a way that the resulting set of answers expresses, as closely as possible, what the users defined as important at query submission time, considering the dynamicity of the environment.

Two aspects should be considered when dealing with query reformulation. First, querying distributed data sources should be useful for users, i.e., resulting query answers should be in conformance with users' preferences. A second aspect is that concepts from a source peer do not always have exact corresponding concepts in a target one, what may result in an empty reformulation and, possibly, no answer to the user. In this sense, we present a query reformulation tool - named *SemRef*, which uses semantics as a way to better deal with these mentioned aspects. The contributions of our tool are twofold: (i) in order to capture user preferences, query semantics and environment parameters, we use *context*, i.e., the circumstantial elements that make a situation unique and comprehensible [Dey 2001]; and (ii) we accomplish query reformulation and adaptation through *query enrichment*, by using, besides equivalence, other correspondences which go beyond the ones commonly found (e.g., aggregation and closeness). Through this set of semantic correspondences, and, taking into account the context, we may produce two kinds of query reformulations: (i) an *exact* one,

considering equivalence correspondences and (ii) an *enriched* one, resulting from the set of the other correspondences. As a result, users are provided with a set of expanded answers, according to their preferences.

This paper is organized as follows: Section 2 presents an overview of our approach; Section 3 shows our tool through an illustrative example. Section 4 draws our conclusions and points out some future work.

2. The *SemRef* Approach

In a dynamic environment, semantics may be identified considering the user's and/or peers' perspectives or even the query formulation. In our setting, we use ontologies as uniform representations of peer schemas. The peers are grouped within the same knowledge domain (e.g., *Education*), what enables us to use domain ontologies (DO) as background knowledge to identify correspondences between the peer ontologies.

We have defined seven kinds of semantic correspondences [Souza et al. 2009] which were formalized using a notation based on Distributed Description Logics (DDL) [Borgida and Serafini 2003]. Considering two peer ontologies O_1 and O_2, the correspondences between their elements may be of the following types: *isEquivalentTo,* denoted as O_1:x $\equiv$ O_2:y, *isSubConceptOf*, denoted as O_1:x $\sqsubseteq$ O_2:y, *isSuperConceptOf,* denoted as O_1:x $\sqsupseteq$ O_2:y, *isPartOf* denoted as O_1:x $\rhd$ O_2:y, *isWholeOf* , denoted as O_1:x $\lhd$ O_2:y, *isCloseTo* denoted as O_1:x $\approx$ O_2:y, and *isDisjointWith*, denoted as O_1:x $\perp$ O_2:y.

Another kind of semantic knowledge we use is context [Dey 2001]. We use three types of context: (i) of the user, through the set of preferences that they choose; (ii) of the query, through its semantics and the way the query will be reformulated; and (iii) of the environment, where the submission peer, target peers (i.e., to where the query will be reformulated and routed) are identified. Most contextual information used in this work is acquired at query submission time. We have represented such information by a context ontology [Souza et al. 2008]. However, in this paper, we only deal with the context acquired from the user preferences and from the query.

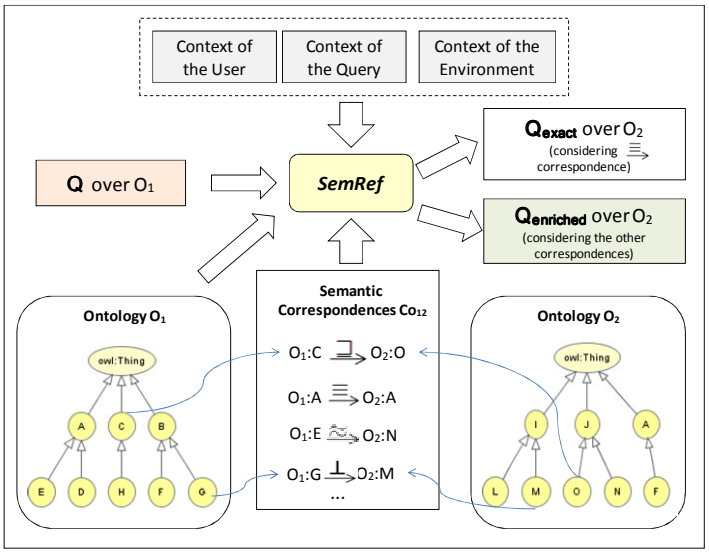

Figure 1. The *SemRef* Approach

Considering these semantic elements, the principle of our approach is to enhance query reformulation by using them in such a way that we can provide users with a set of expanded answers. As depicted in Figure 1, when a query Q is submitted in peer P_1, *SemRef* considers the semantic correspondences (Co_{12}) between the source and target ontologies (O_1 and O_2) along with the concerned context and produces two types of reformulations: Q_{exact} and $Q_{enriched}$. These query reformulations are produced as a means to obtain expanded answers. Since our approach has been encoded in ALC/DL [Baader et al. 2003], we work with queries Q_i composed by disjunctions of queries which are themselves conjunctions of ALC concepts $C_1, \ldots, C_n$ where $n \geq 1$. A query example following such definition is $Q = [Teacher \sqcap Researcher] \sqcup [Student \sqcap Researcher]$ which asks for people who are teachers and researchers or students that are also researchers.

In Figure 2, we present a use case diagram which shows the functional requirements that have been considered in the *SemRef*'s implementation. There are four actors in the diagram. The first actor is the *User* who may define his preferences in terms of *enriching* variables, *path_length* parameter (i.e., the number of subsequent reformulations in the set of neighbor peers), and query reformulation *mode* (i.e., *restricted* or *expanded)*. User preferences are stored as contextual elements to be later verified by the *query reformulator*. The second actor is the *Query Handler* which is responsible for analyzing the query semantics (e.g., required entities and operators) and for receiving and integrating query answers from remote peers. The third actor – *Query Reformulator* - is the main module of the *SemRef* tool. It verifies the acquired contextual elements and existing semantic correspondences between source and target peers and reformulates the query producing Q_{exact} and/or $Q_{enriched}$. For performance reasons, it puts both reformulations together in one execution query (Q') and sends it to the target peer. The fourth actor is the *administrator* who can check whether the reformulation has been done correctly and query answers have been produced accordingly through logs. In next section, we provide some implementation issues and present the tool through an example.

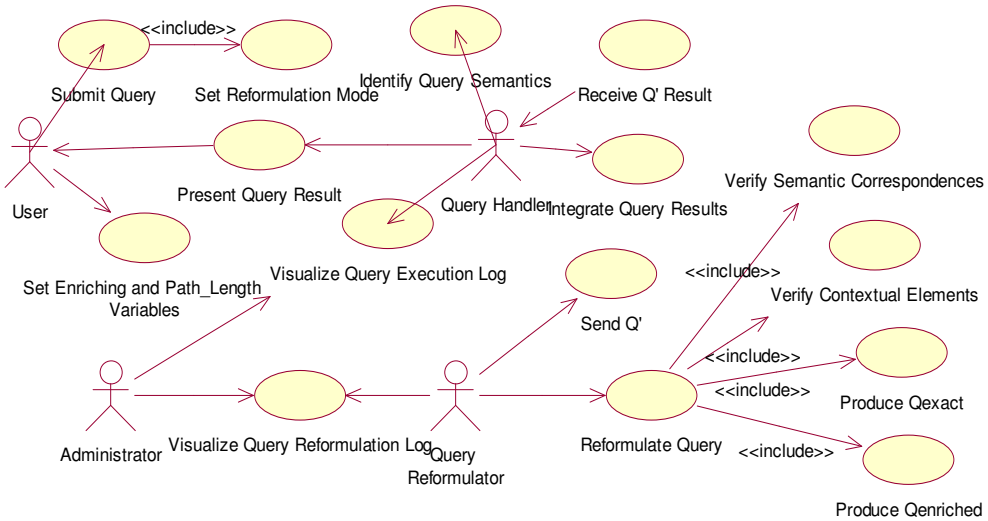

Figure 2. Use Case Diagram for *SemRef*

3. The *SemRef* Tool

The *SemRef* tool has been implemented in Java, and RMI[1] has been used for peer communication. We have adopted both Jena[2] and Protégé's API[3] in order to manipulate the underlying ontologies and execute queries over them. Figure 3 shows a screenshot of the tool's main window that is split into three parts: (i) the peer ontology area; (ii) the query formulation area and (iii) the query results area. Queries can be formulated using the concepts provided by the peer ontology, using SPARQL[4] or using ALC/DL. In this version, we have implemented both ALC/DL and SPARQL options. The reasons underlying these choices were that it was important to validate our approach using ALC/DL, since it has been formally coded as such, and we execute queries over ontologies that represent data sources. Thus, we decided to use an ontology query language. Due to the fact that SPARQL is the W3C proposed standard, it has been chosen as such. However, since SPARQL is composed by a broad range of constructors and query formats, we have defined some templates which may be used by users to write their SPARQL queries in the same way they would do in ALC/DL. The templates are displayed near the corresponding query formulation area, as depicted in Figure 4.

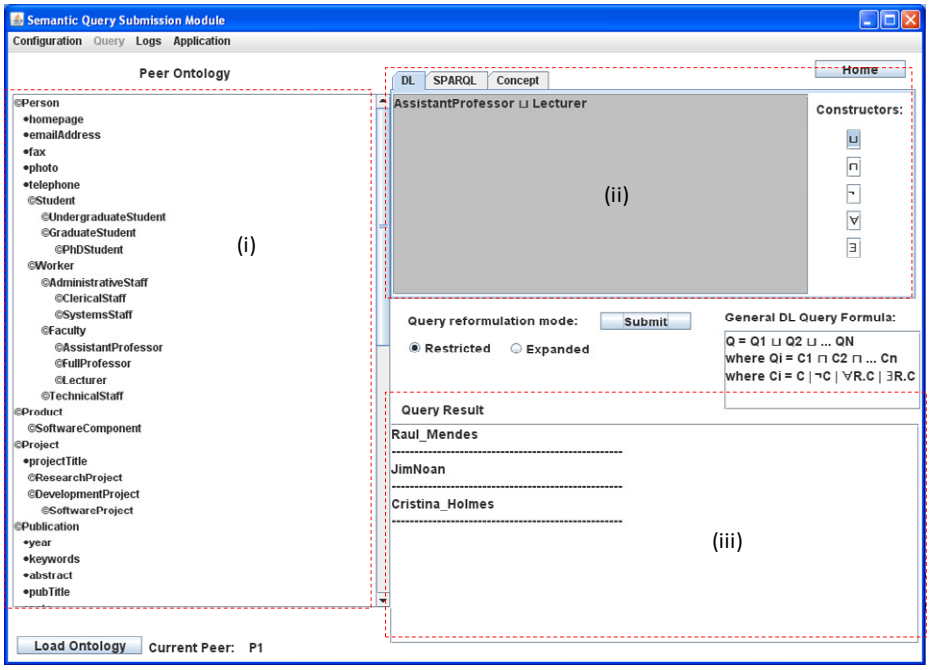

Figure 3. Query Interface with ALC/DL Query Formulation Option

In order to present the tool, we use a scenario composed by two peers P_1 and P_2 which belong to the "Education" knowledge domain. Each peer is described by one ontology – O_1 (*Semiport.owl*) and O_2 (*UnivBench.owl*). We have considered as background knowledge a public DO named *UnivCSCMO.owl*[5]. The set of semantic correspondences between O_1 and O_2 was identified and stored in a RDF file. A fragment of such file (concerning the concept AssistantProfessor) is shown in Figure 5.

[1] http://java.sun.com/j2se/1.4.2/docs/guide/rmi/
[2] http://jena.sourceforge.net/
[3] http://protege.stanford.edu/
[4] http://www.w3.org/TR/rdf-sparql-query/
[5] The complete ontologies are available at http://www.cin.ufpe.br/~speed/ontologies/Ontologies.html

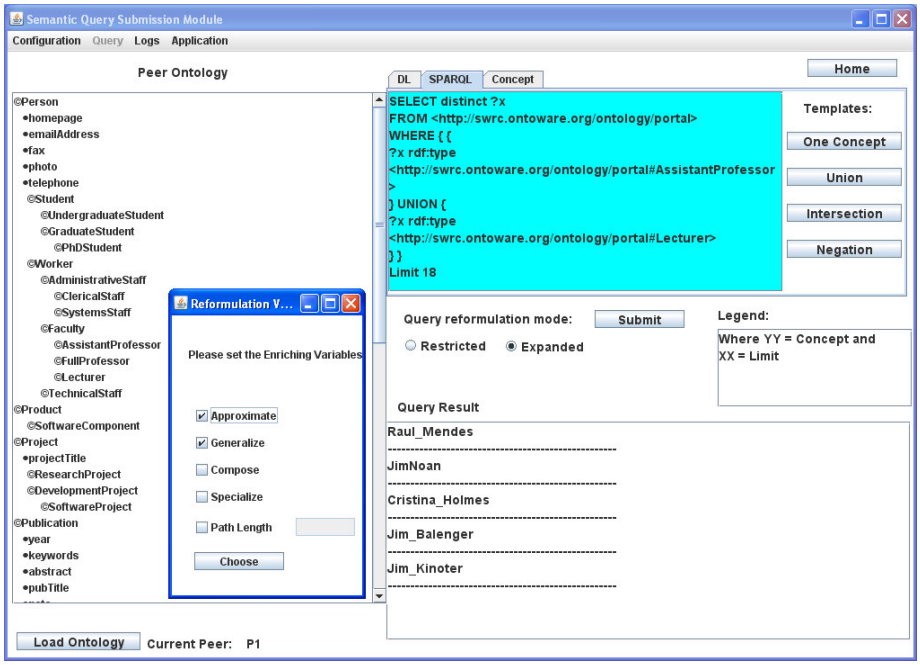

Figure 4. Query interface with SPARQL Query Formulation Option

```
<rdf:Description rdf:about="http://swrc.ontoware.org/ontology/portal#AssistantProfessor">
   <j.0:isCloseTo>http://www.lehigh.edu/~zhp2/univ-bench.owl#VisitingProfessor</j.0:isCloseTo>
   <j.0:isDisjointWith>http://www.lehigh.edu/~zhp2/univ-bench.owl#AssociateProfessor</j.0:isDisjointWith>
   <j.0:isSubConceptOf>http://www.lehigh.edu/~zhp2/univ-bench.owl#Professor</j.0:isSubConceptOf>
 </rdf:Description>
```

Figure 5. Some Correspondences between P_1 and P_2

As an illustration, consider the following ALC/DL Q = AssistantProfessor ⊔ Lecturer (Figure 3) which has been submitted using the *restricted* query reformulation mode option, and *no* enriching variables. As a result, Q_{exact} was empty and the query answers presented in Figure 3 belonged only to the source peer. Then, the same query was submitted in SPARQL (Figure 4). In this case, the *approximate* and *generalize* variables were chosen. Also, the *expanded* reformulation mode was set. Thereby, now, the *SemRef* was able to produce an enriched reformulation which was executed in the target peer. At end, expanded answers from the source and target peers were integrated and displayed in the interface (in the query results area).

The tool also provides two logs: a reformulation log and a query results log. The former shows detailed information about the reformulation process, and the latter shows information about query execution and produced results. Besides the original query, the log presents the produced query reformulations as well as the chosen user preferences. Figure 6 shows a fragment of the Reformulation Log for our query example Q, when it was submitted in expanded mode, with two enriching variables stated by the user .

```
Query Reformulation Mode: Expanded
Using Enriching Variables: Yes
Selected Variables: Approximate – Generalize
Original Query (Source Peer): AssistantProfessor ⊔ Lecturer
Exact Query (Target Peer):
Enriched Query (Target Peer): [[VisitingProfessor ⊔ Professor]] ⊔ [[PostDoc ⊔ Professor ⊔ Faculty]]
```

Figure 6. Reformulation Log for Q submitted in ALC/DL

4. Conclusions and Future Work

In environments which are highly dynamic, the semantics surrounding queries are rather important to produce results with relevance according to users' needs and environment's capabilities. This work has presented a semantic-based query reformulation tool that brings together both query enrichment and query reformulation. The *SemRef* tool has put the theoretical foundations we have provided in Souza et al. [2009] in practice. Through our implementation solution, we provided users with queries in ALC/DL and SPARQL. To this end, we have bridged the gap between ALC/DL semantics in terms of SPARQL, by creating some templates that match each ALC/DL constructor. In order to facilitate query formulation, we have designed the interface in such a way that users use patterns both to ALC/DL and SPARQL options. We have also created logs which show how query reformulation was performed as well as query answers have been produced. As a result, administrators can verify the correctness and adequacy of both tasks.

Currently, we are developing rules to allow reasoning over the instantiated contextual information (of the query, of the user and of the environment). This reasoning might improve the query reformulation and routing processes. As further work, we will instantiate additional query reformulation scenarios which may allow us to work with other different contextual settings and with larger datasets.

References

Adjiman, P., Goasdoué, F., Rousset, M.-C. (2007) "SomeRDFS in the Semantic Web", In Journal on Data Semantics, LNCS, 2007, vol. 8, pp. 158-181.

Baader, F., Calvanese, D., McGuinness, D., Nardi D., and Patel-Schneider P. editors (2003) "The Description Logic Handbook: Theory, Implementation and Applications". Cambridge University Press.

Borgida A. and Serafini L. (2003) "Distributed description logics: Assimilating information from peer sources", Journal of Data Semantics. LNCS 2800, Springer Verlag, pp. 153–184.

Dey, A. (2001) "Understanding and Using Context", Personal and Ubiquitous Computing Journal, Volume 5 (1), pp. 4-7.

Necib C. B. and Freytag J. (2005) "Query Processing Using Ontologies". Proceedings of the 17th Conference on Advanced Information Systems Engineering (CAISE'05), Porto, Portugal, 2005.

Kostadinov D. (2007) "Data Personalization: an approach for profile management and query reformulation". PHD Thesis. Universite de Versailles Saint-Quentin-en-Yvelines, 2007.

Souza D., Arruda T., Salgado A. C., Tedesco P., and Kedad, Z. (2009) "Using Semantics to Enhance Query Reformulation in Dynamic Environments". To appear in the Proceedings of the 13th East European Conference on Advances in Databases and Information Systems, Riga, Latvia.

Souza, D., Belian, R., Salgado, A. C., Tedesco, P. (2008) "Towards a Context Ontology to Enhance Data Integration Processes", In Proceedings of the 4th Workshop on Ontologies-based Techniques for DataBases in Information Systems and Knowledge Systems (ODBIS). VLDB '08, Auckland, New Zealand, pp. 49-56.

Stuckenschmidt H., Giunchiglia F., and van Harmelen F. (2005) "Query processing in ontology-based peer-to-peer systems". In V. Tamma, S. Craneeld, T. Finin, and S. Willmott, editors, Ontologies for Agents: Theory and Experiences. Birkhuser. (2005).