

# Improving User Interaction on Ontology-based Peer Data Management Systems

Andreza Leite de Alencar<sup>1,2</sup>, Ana Carolina Salgado<sup>1</sup>

<sup>1</sup>Informatic Center, Federal University of Pernambuco, Recife, Brazil

<sup>2</sup>Deinfo, Federal Rural University of Pernambuco, Recife, Brazil

ala4@cin.ufpe.br, acs@cin.ufpe.br

**Abstract.** *The issue of user interaction for query formulation and execution has been investigated for distributed and dynamic environments, such as Peer Data Management Systems (PDMS). Many of these PDMS are semantic based and composed by data peers, which export schemas that are represented by ontologies. In the literature, we can find some proposed PDMS interfaces, but none of them addresses, in a general way, the needs of a PDMS for user interaction. In this work, we propose a visual user query interface for ontology-based PDMS. It provides a simple and straightforward interaction with this type of system. It aims not only providing a natural visual query interface but also supporting a precise and direct manipulation of the data schemas for query generation, with queries based on ontology browsing and multiple views, graphical network visualization, in addition to user feedback.*

## 1. INTRODUCTION

User interaction for the formulation and execution of queries has been investigated for distributed and dynamic environments, such as Peer Data Management Systems (PDMS) [Blanco et al. 2006]. These environments are dynamic with an infrastructure of data sources (e.g., databases, websites, files), here referred to as *peers*, that are distributed, heterogeneous and autonomous. These peers are linked to each other by means of schema mappings (e.g. semantic associations between schema elements), called hereafter as *correspondences*.

A special problem concerning these architectures is how to exploit the correspondences in order to answer queries and provide users with relevant results. Particularly, the crucial point is how to reformulate queries among the peers 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 current status of the environment. When dealing with query reformulation, two aspects should be considered. First, querying distributed data sources should be useful for users, e.g., resulting query answers should be in conformance with users' preferences. The second aspect is that terms from a source peer do not always have exact corresponding terms in the target one. It may result in an empty reformulation and, possibly, no answer to the user. Regarding the former aspect, we argue that user preferences and the current status of the environment should be taken into account at query reformulation time; regarding the latter, the original query should be adapted to bridge the gap between the two sets of terms.

In this work, we extend a query reformulation approach named SemRef [Souza 2009], which uses semantics as a way to better deal with the aforementioned

aspects. The priority is to produce the best query reformulation through equivalence correspondences, but if that is not possible, or if users define that it is relevant for them to receive semantically related answers, an enriched reformulation is also generated. As a result, users are provided with a set of expanded answers, according to their preferences. The system allows users to see detailed information, when it shows the results, such as the semantic correspondences used by the reformulation process to generate the results. This way, the system allows the users to indicate their satisfaction with the enriched results through an explicit feedback, accepting or rejecting such correspondences. This feedback is saved and considered by the system to enhance query reformulation process in future query submissions.

The existing PDMS query interfaces do not address the needs of a PDMS interface regarding user interaction [Adjiman et al. 2007, Castano et al. 2003, Mandreoli et al. 2008, Montanelli and Castano 2008, Athanasis et al. 2004, Franconi et al. 2010, Beneventano et al. 2007]. Among the existing approaches, all of them include the formulation and submission of queries but they do not offer these features as a visual interface, e.g., with queries based on ontology browsing and multiple views, graphical visualization of the network, in addition to user feedback.

The proposed interface introduces the visualization of ontologies terms as a graph or as a hierarchy. However, it provides a visual query formulation by ontology manipulation, by terms selection and their combination through a set of constructors available in the interface. Another way to submit queries is by using SPARQL Protocol and RDF Query Language (SPARQL) [Prud'hommeaux and Seaborne 2008]. SPARQL templates are available in our interface and the users just need to insert the terms (or concept) to be queried. Likewise, the interface offer the possibility to improve the results by enriching the submitted query through the use of variables that represent relationships between the query's term and related terms in the corresponding domain ontology (e.g., approximation, sub-concept, super-concept and aggregation) . In addition to query features, our interface allows an organized view of the results. Also, such results include information about the semantic correspondences used to enrich the query. In addition, a graphical visualization of the network topology highlights the peers that participated to produce results.

The interface was developed and validated in a PDMS prototype named SPEED (Semantic PEER-to-Peer Data Management System) [Pires 2009] but it is important to highlight that the specification of the interface was defined based on PDMS needs and so can be applied for any PDMS.

The paper is organized as follows: Section 2 presents the main PDMS concepts and describes SPEED (Semantic PEER-to-Peer Data Management System); Section 3 presents our approach of a visual query interface with the specification and implementation; Section 4 presents the results and the evaluation of the interface; Section 5 presents some related work and compares them to our approach; Section 6 presents the conclusions and future work.

## **2. PEER DATA MANAGEMENT SYSTEMS**

Peer Data Management Systems (PDMS) [Halevy et al. 2006, Mandreoli et al. 2007, Mandreoli et al. 2008] came into the focus of research as a natural extension to distributed

databases in a Peer-to-Peer (P2P) setting [Herschel and Heese 2005]. They consist of a set of peers, each one with an associated schema (called exported schema) that represents the data shared with other peers. In such systems, schema matching techniques are used to establish semantic correspondences which form the basis for query answering and peer clustering. The semantic correspondences' definition occurs among pairs of semantic neighbour peers, e.g., peers that are semantically related and previously identified by a peer clustering process. Query results submitted at a peer is composed by data residing at that peer and data that is reached through correspondences over the semantic neighbours.

In the overlay network, if peers that are semantically similar are put together, it can improve semantic correspondences' establishment and, thereafter, query answering in PDMS. In this sense, Pires [Pires 2009] proposed a semantic-based PDMS, named SPEED (Semantic PEER-to-Peer Data Management System), whose mixed network is mainly designed to assist the organization of peers according to their exported schema (represented by an ontology).

PDMS are highly dynamic systems. To help matters, Ontology-based PDMS uses semantics as a way to enhance tasks such as peer clustering, and, particularly, query reformulation [Arruda et al. 2010, Souza et al. 2009]. In these systems, the peers are clustered according to the same knowledge domain, and an ontology describing the domain is available and used as background knowledge. Ontologies are also used to a uniform conceptual representation of peer's schemas and correspondences between these ontologies are set to provide their data sources' background.

## **2.1. The SPEED System**

In this section, we describe the PDMS SPEED [Pires 2009]. The system adopts a mixed P2P network topology: DHT [Stojanovic et al. 2004], super-peer [Yang and Garcia-Molina 2003], and unstructured [Freenet 2014]. In order to assist peer organization in the network according to their exported schemas, an exploration of the strengths of such topologies occurs. SPEED's main goal is to cluster semantically similar peers in order to facilitate the establishment of semantic correspondences among peers and, thereafter, improve query processing on a large number of data sources [Pires 2009]. Next, we present an overview of SPEED's architecture and query process.

In SPEED, ontologies are used as an uniform conceptual representation of peer schemas and schema matching techniques are used to establish semantic correspondences that form the basis for query answering and peer clustering processes [Pereira et al. 2011]. The definition of semantic correspondences occurs amid pairs of semantic neighbour peers, e.g., peers that are semantically related and previously identified by a clustering process.

### **The SPEED's Architecture**

A DHT network is used to link particular peers representing a certain knowledge domain. Peers are grouped according to their knowledge domain (e.g., Education and Health), forming semantic communities. Inside the community peers are organized in a finer grouping level, named semantic clusters, where peers share similar ontologies (schemas). A semantic cluster has a cluster ontology, which represents the ontologies (schemas) of the peers within the cluster. Each cluster maintains a link to its semantic neighbours in the overlay network, e.g., for other semantically similar clusters. In other words, semantically

similar peers form clusters in a super-peer network considering their exported schemas. As illustrated in Figure 1, three distinct types of peers are considered in the proposed system: data peers ( $I_n D_n$ ), integration peers ( $I_n$ ), and semantic peers ( $S_n$ ).

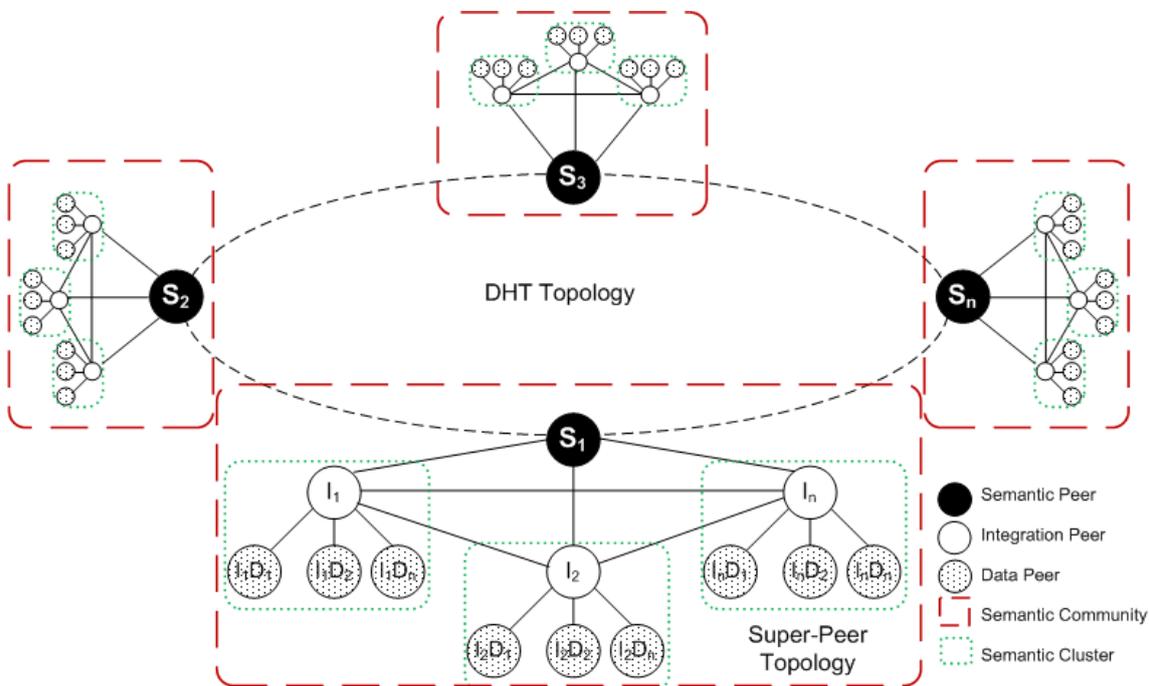


Figure 1. SPEED's Architecture (adapted from [Pires 2009]).

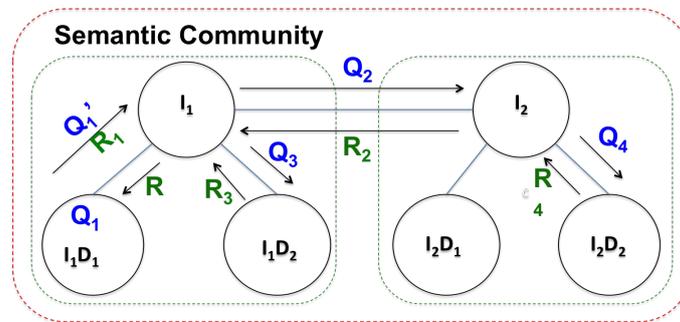
A data peer is a simple computer or a server storing an autonomous data source.  $I_1 D_1$ ,  $I_1 D_2$ , and  $I_1 D_n$  are examples of data peers. Data peers are logically organized in a super-peer network. In this sense, semantic clusters are formed according to data peers' exported schema. Each semantic cluster has a special type of peer named integration peer. In fact, an integration peer is a data peer with higher computational capacity. It is responsible to define and to maintain semantic correspondences as well as for managing query processing and data peer's metadata. For instance,  $I_1$  is the integration peer of the cluster formed by the data peers  $I_1 D_1$ ,  $I_1 D_2$ , and  $I_1 D_n$ . A set of clusters sharing content belonging to a common knowledge domain forms a semantic community. Each semantic community has a special type of peer named semantic peer.  $S_1$  is an example of a semantic peer. Connections between semantic peers are a DHT network while connections between integration peers are an unstructured network. Our approach assumes that an integration peer names its respective cluster while a semantic peer names its corresponding semantic community (e.g. semantic cluster  $I_1$  and community  $S_1$ ). The clusters  $I_1$ ,  $I_2$ , and  $I_n$  form the semantic community  $S_1$ .

### Query Processing

In SPEED, queries can be posed and answered by data peers and integration peers. They are formulated according to the peer's exported schema and internally translated into the common query model. A query is disseminated only among the clusters of the semantic community that originated the query. Semantic peers do not participate in the query processing. Thereafter, if a query is submitted at a particular semantic community the query is forwarded to other communities. During the navigation, a query is

reformulated according to previously established semantic correspondences. Integration peers are responsible for integrating query results received from its data peers and other integration peers [Pires 2009, Arruda et al. 2010, Souza et al. 2009].

To better explain how query processing occurs in SPEED, consider again the setting described in Figure 2 containing one semantic community and two clusters. Assume that queries and exported schemas are represented in a common data model. Suppose that query  $Q_1$  is submitted at the data peer  $I_1D_1$  (Figure 2). In this case,  $Q_1$  is translated into a query  $Q_1'$  described in the common query model. Afterwards,  $Q_1'$  is sent to its corresponding integration peer  $I_1$  in order to be propagated in the semantic community. In parallel,  $Q_1$  is executed at the data peer  $I_1D_1$ , and the query result ( $R_1$ ) is sent to the integration peer  $I_1$  to the integration with other query results possibly received from other peers.



**Figure 2.** Query processing in the SPEED environment (adapted from [Pires 2009]).

Once the integration peer  $I_1$  receives  $Q_1'$ , it uses its semantic correspondences to identify relevant peers for  $Q_1'$ . In the example, assume that the relevant peers are the data peer  $I_1D_2$  as well as the integration peers  $I_1$  and  $I_2$ . Thus,  $Q_1'$  is reformulated into the queries  $Q_2$  and  $Q_3$  and forwarded to  $I_2$  and  $I_1D_2$ , respectively. Particularly,  $Q_1'$  is executed at  $I_1$  where the corresponding query result integrates with the ones obtained from  $I_1D_1$ ,  $I_1D_2$ , and  $I_2$ . Assume that  $I_2$  has identified  $I_2D_2$  as a relevant peer to answer the query ( $I_2D_1$  is not relevant). Thus,  $Q_2$  is reformulated into  $Q_4$  and forwarded only to  $I_2D_2$ . When the reformulated query  $Q_3$  arrives at the data peer  $I_1D_2$ , this executes the query  $Q_3$  and the query result ( $R_3$ ) is returned to the integration peer  $I_1$  for data integration. Similarly, the data peer  $I_2D_2$  executes  $Q_4$  in its local data source and returns its query result ( $R_4$ ) to the integration peer  $I_2$ . At  $I_2$ , the query result  $R_4$  is integrated with the one of  $I_2$  and returned to  $I_1$  ( $R_2$ ).

At the integration peer  $I_1$  occurs the integration of the query results  $R_1'$ ,  $R_2$ , and  $R_3$ . Then, the final query result ( $R$ ) is returned to the data peer  $I_1D_1$  where the initial query  $Q_1$  was submitted and, finally, the integrated query results are presented to the user who has formulated  $Q_1$ .

Section 3 presents a flow diagram and examples of use to illustrate the use of the interface and the query processing.

### 3. THE VISUAL QUERY INTERFACE

In this work, we propose a visual query interface, named VisualPDMS, for ontology-based PDMS. It aims at providing a natural visual query interface but it also supporting precise direct manipulation through automated query generation [Li et al. 1997]. This interface was specified, using analysis and prototyping techniques [Dix et al. 2003]. In this section, we present the specification of the interface and describe its main characteristics.

#### VisualPDMS Layers

The features of the interface were defined based on ontology-based PDMS needs. The interface was implemented in the context of the SPEED system to validate the approach, but the definitions can be used for any ontology-based PDMS. To accomplish the task, we have defined two new layers added to SPEED's architecture: (1) a user interaction layer and (2) a management layer. Figure 3 illustrates the proposed new layers and their modules on the solid line and the third layer, which represents SPEED's core, on dotted line.

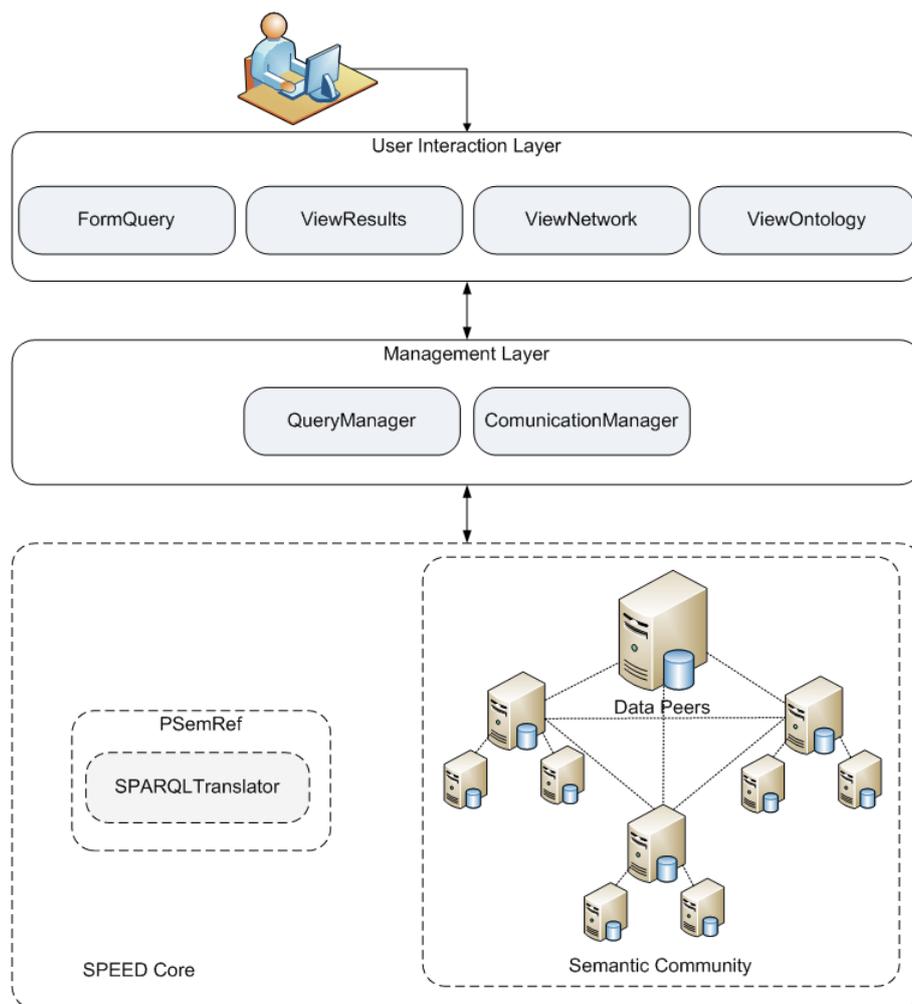


Figure 3. The proposed architecture.

The User Interaction Layer consists of five modules:

- ViewOntology - responsible for converting ontologies, which represent the schemas of data sources, to be visualized as a graph or as a hierarchy of terms. These options will be available as a user choice according to his/her preference for visualizing and browsing the ontology.
- FormQuery- responsible for formulating queries that are sent to the query module (PSemRef). This formulation addresses the need for various types of user and can be done in two ways: (1) for users participating in the network, based on direct visualization and manipulation of the local data schema, (2) for users not connected to participant peers, through the choice of a domain and a starting peer, which will be informed by the system, followed by visualization and manipulation of the data schema of the selected peer. These options for formulating queries will be available automatically in accordance with the type of user who is interacting with the system.
- ViewResults - responsible for organizing and displaying the results of queries. This module is also responsible for displaying detailed information about the results, such as the semantic correspondences that generated them and the submission peer. A graphical visualization of the network topology is provided.
- ViewNetwork - responsible for displaying the network topology enabling the user to know the neighbours of the peer in which s/he is interacting.

The Management Layer has two modules responsible for the communication between the User Interaction and the PDMS core layers:

- QueryManager - responsible for managing the queries formulated by the user in the FormQuery module so that they can be submitted to the PSemRef module. In addition, it performs the integration of query results translating them into a format comprehensible by the ViewResults module.
- CommunicationManager - responsible for communicating the User Interaction Layer with PDMS core. This module acts as a proxy between the user interaction modules and all other PDMS components required for interaction features.

SPEED's core contains relevant components that are not in the scope of this work, but can help to illustrate the features:

- PSemRef - responsible for processing and reformulating queries to the other peers in the community [Arruda et al. 2010].
- Semantic Community - represents groups of data peers (semantic clusters) according to their semantic interests, e.g., their domain of interest and local ontology.
- Data Peers - in general terms, represents the data sources to be queried by the system.

An analysis of the architecture helped to state some desirable features and functionalities to be offered by the interface. First, to the query submission, the system must offer a user-friendly interface and must be comprehensible to the queries' formulation and to the result visualization. Second, different types of users need an interface tailored to their needs and preferences. Third, it is desirable to have a graphical interface for visualizing the network topology, enabling access to the information available on it, as well as a registration form that allows the entry of new data peers. The interface was prototyped based on the analysis of requirements; architecture and

diagrams defined in the specification process. For the interface, we have considered two different types of users: the participant user in the PDMS system (PU, for short) and the non-participant user in the system (NPU).

To illustrate the query process, Figure 4 shows a twelve steps flow diagram that also illustrates the communication among the involved modules of the architecture.

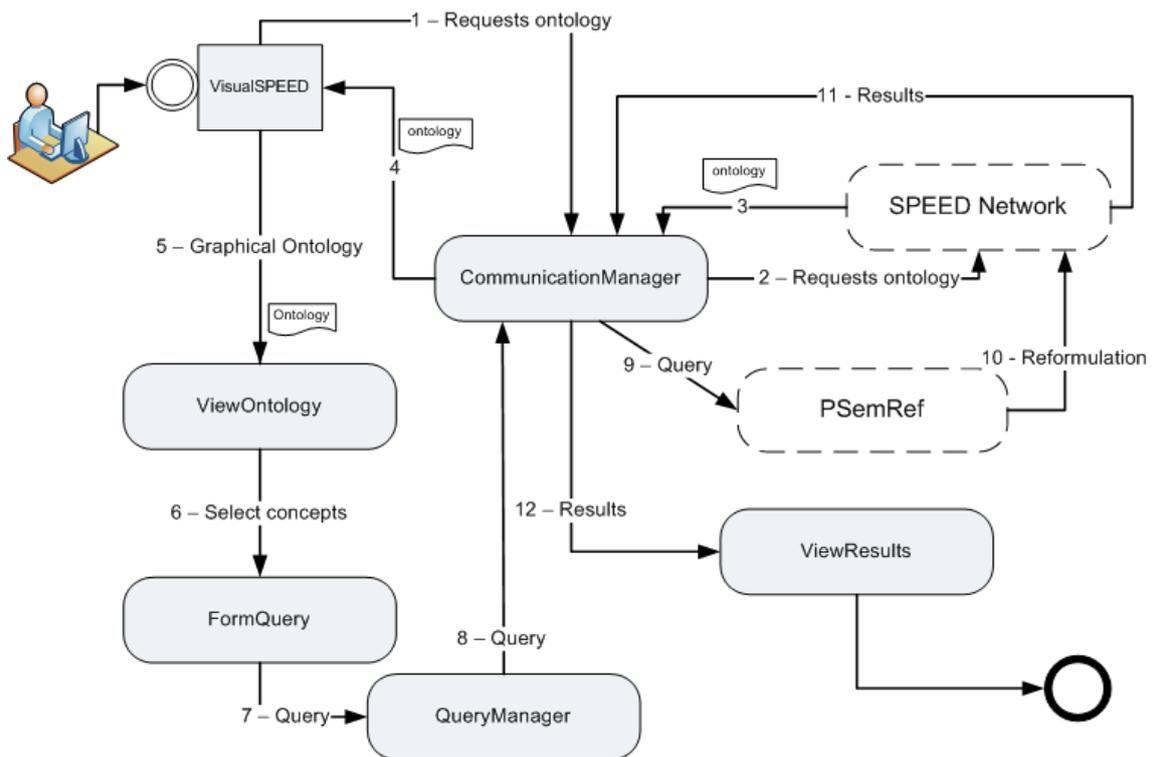


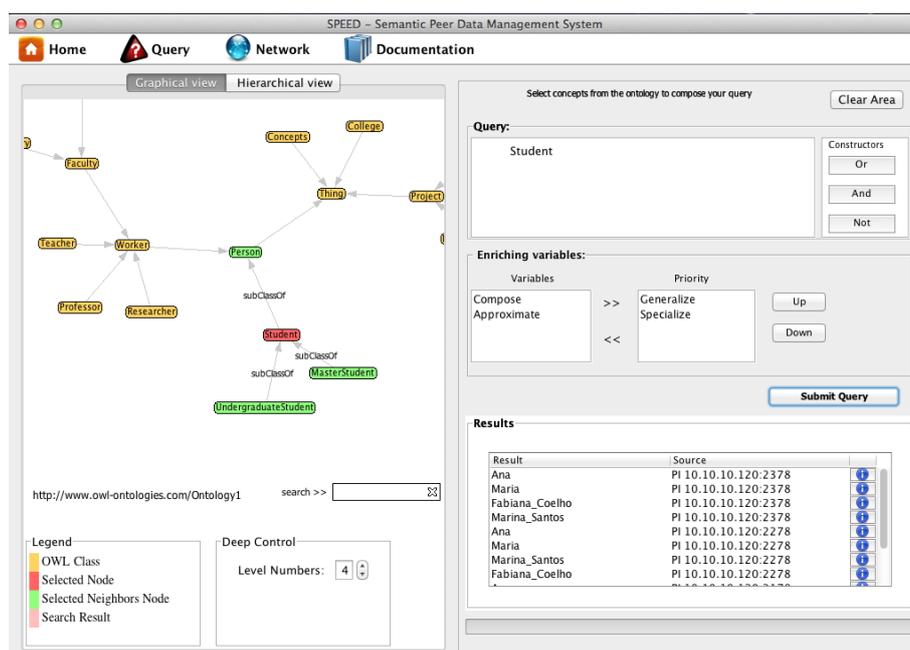
Figure 4. Query flow diagram.

### Implementation and Use

The interface was implemented as part of the PDMS SPEED [Pires 2009] to validate the approach. All modules were implemented in the Java programming language [Java 2014], along with Integrated Development Environment (IDE) Eclipse [Eclipse 2014].

The system will provide for both types users four items in the menu: Home, Query, Network and Documentation. Home and Documentation are available as default for both user types. The Home menu displays detailed information as the description of the system and the participating institutions (peers). The Documentation menu will present the documents available to users of the system, for example, a user guide and a Javadoc. The Query menu has different characteristics for the submission of queries for each user. A user PU accesses the system via a data peer or an integration peer. These users can visualize their data schema in the form of a graph or as a hierarchy of terms. Figure 5 shows the query interface with a preview of an ontology shaped as a graph. This graphical visualization allows user to manipulate the ontology emphasizing points of interest by navigating the graphical structure or setting levels of desired depth.

To form a query, the user selects the terms desired in the graphical representation of the ontology. The selected terms will be showed on the query composition field, which



**Figure 5. The proposed query interface.**

is in the upper right side of Figure 5. In this area, constructors as OR, AND or NOT can be used to compose the query. The query submitted by the user is interpreted by the system and translated to a SPARQL command to be executed in a data or integration peers.

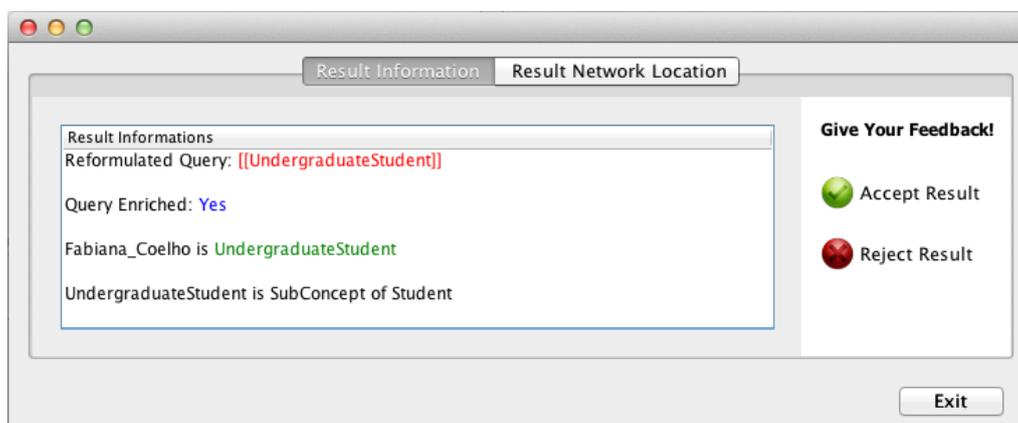
The system offers the possibility to improve the query results by enriching the submitted query. Users can customize the query by selecting and prioritizing enriching variables (approximation, sub-concept, super-concept and aggregation) [Arruda et al. 2010, Souza et al. 2009]. These variables represent semantic relationships between the terms of the query represented by the generated semantic correspondences. Figure 5 shows, in the bottom right side, the enrichment of the query.

Another option to submit queries by expert users is using SPARQL (SPARQL Protocol and RDF Query Language). By selecting the option for submitting SPARQL query, users will have the help of templates since the language SPARQL is not intuitive for most of the users. By choosing the templates, the user needs to insert just the terms to be queried, selecting them from the graphical representation of the ontology. Each SPARQL template has the structure of a particular query.

For the submission of queries by a NPU, as this type of user is not connected to use a data peer in the system network, s/he must connect first to a data peer to have access to his/her data schema and proceed with the formulation of the query. Thus, the system should lead the user from the beginning of the formulation of the query until the results visualization. For this purpose, two stages were defined to the query formulation: (1) Domain selection - in this step the user must choose a domain for submission of queries from a list informed by the system. (2) Starting peer selection - the system will display the participating peers of the semantic community, associated with the chosen domain, and the user must choose a starting peer for submission of queries. After the user select

the starting peer, the NPU can visualize the schema of the chosen peer and proceed with the query following the same pattern used for a PU.

After query execution, the system displays the results organized in a table shown in the bottom right side of Figure 5. This result visualization allows users to see detailed information about the results, such as semantic correspondences used by the reformulation process through data peers. By offering the used semantic correspondences that generated the results, the system allows the user to indicate their satisfaction with the enriched results, accepting or rejecting such correspondences as showed in the right side of Figure 6. Providing the users with the possibility of removing correspondences is a good way to obtain knowledge about incorrect correspondences [Pereira 2011]. This feedback is saved and considered by the system to enhance query reformulation process in future query submissions.



**Figure 6. Result informations (semantic correspondences) and user feedback.**

Users can also see the identification of data peers that originated the result presented as a graphical representation of the network topology. The Network menu also presents the visualization of the network topology.

#### 4. Evaluation and Results

Regarding the assessment of the interface and verification of compliance with usability criteria, examples of use were performed with two invited types of users (experts and non-experts). In this section, we describe the scenario used in the test of the interface. The domain used for user interaction in the tests was *Education*. For these tests, we perform a query  $Q_1 = Student \sqcup Project$ , which seeks projects or students, by selecting the terms in the graphic visualization of the ontology and the constructors available on the interface. In these tests, we consider two ways to submit the same query: (1) without the option of enrichment and (2) using the enrichment variables *Generalize* and *Specialize*, following this order of priority. In addition to the previous submission, we submitted the query  $Q_2$  in SPARQL option using the same terms and the SPARQL template *Union*.

The submitted query  $Q_2$  was as follows.

```
SELECT
  distinct ?x
```

```

FROM
  /www.owl.com/Ontology.owl>
WHERE { {
  ?x rdf:type
  /www.owl.com/Ontology.owl#Student>
} UNION {
  ?x rdf:type
  /www.owl.com/Ontology.owl#Project> } }

```

After submitting these queries ( $Q_1$  and  $Q_2$ ), users could visualize the results and explore the available information, as the semantic correspondences that have generated them and the peer involved in the query process.

### **Evaluation**

Before declaring a software ready to use, it is important to know whether it adequately supports users in their tasks and in the environment in which the software will be used. In addition, functionality tests are needed to verify the robustness of the implementation. Evaluation of usability of the interface is necessary to analyse the software quality of use [Prates and Barbosa 2003, Rogers et al. 2011]. In this work, we have performed two evaluations. One evaluated the functionalities available in the interface. Another one evaluated the usability assuring that the interface is in accordance with expected usability criteria.

#### *Functionalities' Evaluation*

Because the focus of this work is to provide user interaction in a PDMS, it is important to get feedback from users about the features available in the system. For this purpose, we invited forty-eight volunteers to use the interface and evaluating their features. Two groups of users were necessary: (1) twenty-four volunteers considered expert and (2) twenty-four users that are not experts in the use of ontologies and SPARQL language.

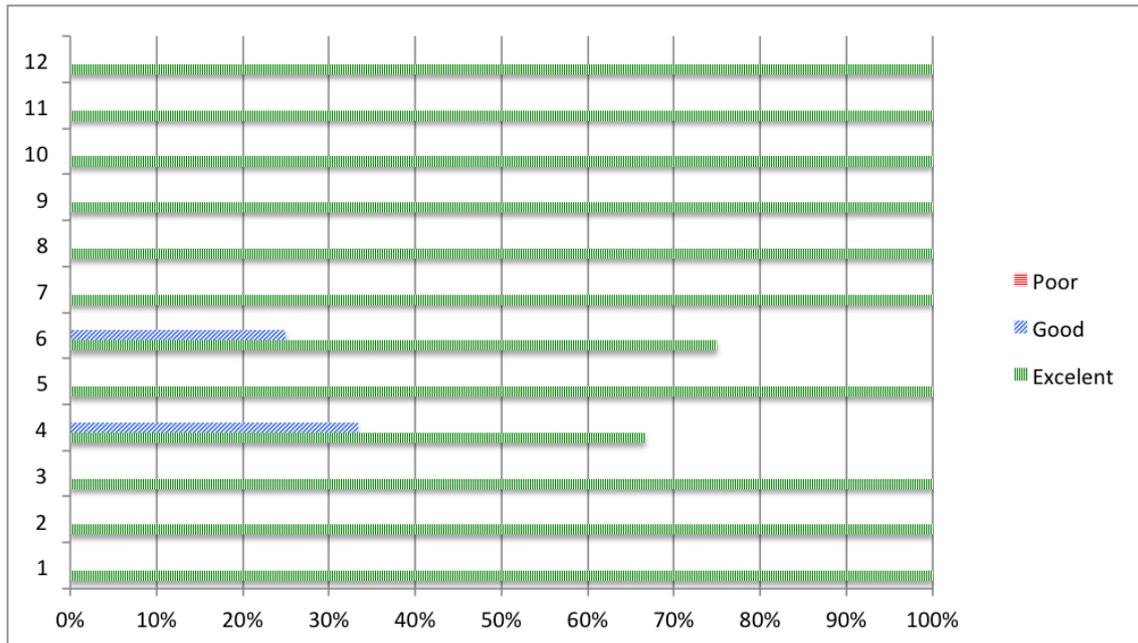
In this sense, we provided to users a script with a step-by-step of the examples of use to be performed. These examples performed the queries  $Q_1$  and  $Q_2$ , submitted in three ways: (1) restricted mode, only with inclusion of terms and constructors, (2) restricted mode, using the SPARQL templates available in the interface, and (3) expanded mode, selecting enrichment variables for the two previous queries. In addition to the queries proposed in the scenario given to users, they also had the freedom to perform any additional query they wanted before giving their opinion about the system.

To get the users' evaluation we have used a questionnaire. This questionnaire consists in twelve questions about the usefulness and relevance of the features available to the user interaction with the system, listed as follows.

1. Domain and peer selection.
2. A sign up form for a new peer entering the system.
3. Display of ontology as a graph.
4. Display of ontology as a terms hierarchy.
5. Visualization of the network topology.
6. Composition of queries and use of constructors.
7. Composition of queries using SPARQL templates.
8. Use and prioritization of enrichment variables.

9. Visualization of query results.
10. Visualization of semantic information of the results.
11. Feedback on the results concerning the semantic correspondences used.
12. Peers' identification where the query was executed.

The graphics illustrated in Figures 7 and 8 display the results of the evaluation by expert users and non-expert users, respectively. The graphics report the percentage of responses (Poor, Good and Excellent) for each question about systems' functionality.



**Figure 7. Results of experts' evaluation.**

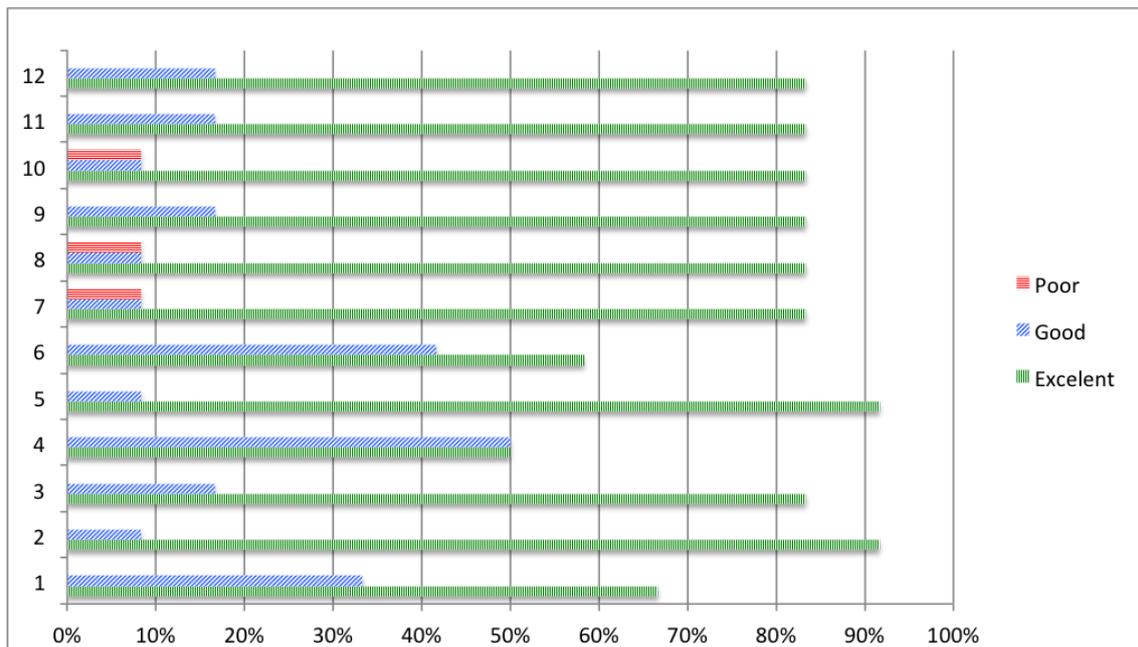
We observe, in the graphic, the majority of users (experts and non-experts) answered a questionnaire indicating that they were satisfied with the features available in the system. In most of the questions, we got answers between Good and Excellent. They confirm the user interface provides a simple and straightforward interaction with the system.

Few users (non-experts) rated negative features such as submission of SPARQL queries, the use of enrichment variables and the visualization of semantic information of query results. In their comments, they justified the answers by the fact that these features do not have much importance or significance to them. However, it was possible to use them due to the facility provided by the interface. Whereas, this type of user has no knowledge about ontologies and SPARQL query language, we do not consider these responses as negative reviews, but as a proof that we reached the goal of promoting a simple and objective interaction regardless the type of user.

### *Ergonomic Evaluation*

To lead an inspection of the ergonomic quality of the user interface, we used the web service *ErgoList* which proposes usability inspections by *check lists* [Cybis et al. 2003].

A variety of professionals, not necessarily experts in ergonomics, for example, programmers and analysts, can use the usability inspections based on checklists to



**Figure 8. Results of non-experts' evaluation.**

diagnoses general and repetitive problems of interfaces [Jeffries et al. 1991].

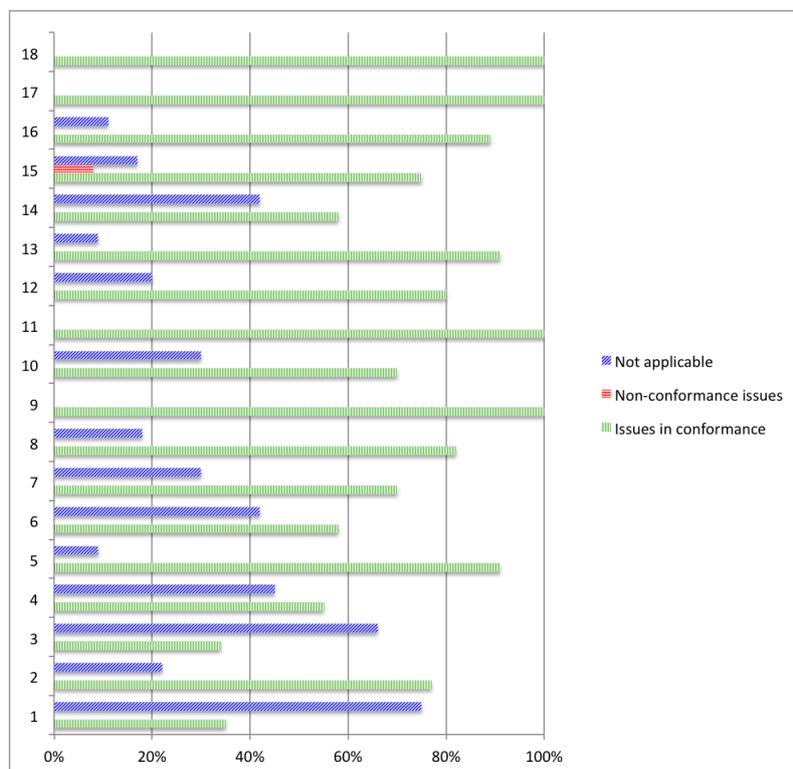
Dominique Scapin conducted a study [Scapin 1990] concerning the organization of knowledge about ergonomics of human-computer interfaces, in order to make them easily available, for both specialists and non-specialists. The criteria system defined by Scapin aims to facilitate the recovery of ergonomic knowledge.

There are eighteen basic criteria to determine the ergonomics of a human-computer interface [LabUtil 2014]:

1. Promptness;
2. Location Format Grouping;
3. Format Grouping;
4. Instant Feedback;
5. Readability;
6. Conciseness;
7. Minimal Actions;
8. Informational Density;
9. Explicit User Minimal Actions;
10. User Control;
11. Flexibility;
12. Consideration of the User Experience;
13. Protection against errors;
14. Quality of Error Messages;
15. Correction of Errors;
16. Consistency;
17. Meanings; and
18. Compatibility.

Figure 9 illustrates the percentage compliance of the proposed interface for each of the criterion evaluated in the checklists. For the construction of the graphic, the final report issued by the *Web Ergolist* was used. It counts the answers of the questions: questions in conformance (ergonomic issues addressed by the proposed interface); in non-conformance (ergonomic issues that the proposed interface does not meet properly); and non-applicable questions (questions that address specific ergonomic not necessary for the proposed interface).

Figure 9 shows the FeedBack criteria that presents a rate of non-compliance. It is because the system does not provide a historical of the commands used by the user when s/he is interacting with this system. For other criteria, besides the percentage of compliance, there is also the percentage of questions not applicable, for instance, questions about presentation and use of numeric values do not apply to the proposed interface.

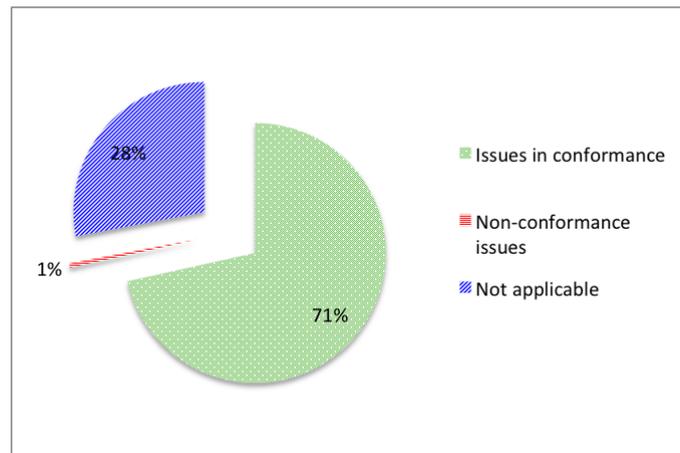


**Figure 9. Percentage of individual adequacy to the *ErgoList* usability criteria.**

In addition to the chart with the percentages for each criterion considered in the evaluation, we designed a chart that illustrates the overall assessment of the interface according to the total compliance reported in the final report issued by the *Web Ergolist*. Figure 10 shows that the interface is in accordance to the usability criteria expected by *ErgoList*.

## 5. RELATED WORK

Several authors [Adjiman et al. 2007, Castano et al. 2003, Mandreoli et al. 2008, Montanelli and Castano 2008] present approaches for processing queries on



**Figure 10. Overall adequacy of the proposed interface.**

ontology-based PDMS. Among those who proposed user interfaces in the formulation and submission of queries, we classified them based on the methodology or technique used to formulate queries: queries based on ontology browsing and queries based on multiple views [Hoang and Tjoa 2006].

Some approaches proposed graphical user interfaces that present a pattern for building the query based on ontology browsing where the user select a class and, by navigation, s/he can specify the artifacts for queries among the relevant properties of the selected class. As an example of this kind of interface we quote the Query Tool [Franconi et al. 2010], Graphical RQL [Athanasios et al. 2004] and SEWASIE [Beneventano et al. 2007].

A query based on multiple views [Makela et al. 2004] combined with the use of ontologies has proved to be a powerful search paradigm [Hyvonen and Viljanen 2003]. In this approach, many different views are provided for the data. Projection of ontologies creates views using several other hierarchical relationships inherent of the ontology. It is the case of Ontogator [Hyvonen and Viljanen 2003] and OntoViews [Makela et al. 2004].

In addition to the approaches found for ontology-based query interfaces, we have found an approach named SUNRISE [Mandreoli et al. 2007], which allows the user to explore the most promising way to route queries in a PDMS network.

Among the approaches considered, all of them include the formulation and submission of queries but do not offer these features as a visual interface. Table 1 presents a comparison between the analysed approaches and our proposal.

The characteristics considered to the comparison were described as follows:

1. Query Formulation - Approaches that proposed techniques for formulating and submitting queries;
2. Results Visualization - Approaches that proposed techniques to results visualization;
3. Network Visualization - Approaches that proposed visualization of the network topology;
4. Implementation - Approaches that has an implementation of the proposed approach;

5. Types of User - Approaches that enable interaction of various user types; and
6. User Feedback - Approaches that enable users to give feedbacks about query results.

**Table 1. Comparison to the existing approaches.**

Approaches	1	2	3	4	5	6
Query Tool	X	-	-	-	X	-
Graphical RQL	X	-	-	-	-	-
SEWASIE	X	-	-	X	-	-
Ontogator	X	X	-	X	-	-
OntoViews	X	-	-	X	-	-
SUNRISE	X	-	X	X	-	-
VisualPDMS	X	X	X	X	X	X

By analysing the table, we can observe that none of the approaches address, in general, all the features and functionalities required for user interaction with PDMS as VisualPDMS does.

## 6. DISCUSSIONS AND FINAL REMARKS

As mentioned before, finding relevant query results is challenging because it is hard to express the user's specific intention in a given query interface, and a keyword query typically retrieves a large number of results. To make use of visual interfaces and user feedback is a functional option to enhance the queries and get relevant results.

This paper proposed a visual interface that provides the functionality and transparency necessary for user interaction with a PDMS. In this context, the contribution of this paper are summarized as follows:

- The definition of a PDMS interface that enables the interaction of various types of users, which provides query formulation in a visual way by manipulating the graphical visualization of ontologies;
- The usability improvement of query constructors, SPARQL instructions and enrichment variables to query composition;
- The visualization of query results in an organized way allowing access to detailed information about them;
- The use of user feedback to enhance query reformulation process; and
- The implementation and evaluation of the proposed interface.

### Future Work

The interface specification was developed to satisfy any PDMS needs, but there is a coupling between the implemented prototype and SPEED's environment and requires some modifications to run in other PDMS environments. Hence, the next step is to develop a multi PDMS interface that offers parametrization and simple updates. In addition, the approach deserves some extensions and improvements, such as the development of a help and user guide, and the execution and assessment in a real environment with a larger number of peers running on different networks and connected through the internet.

## References

- Adjiman, P., Goasdoue, F., and Rousset, M. (2007). Somerdfs in the semantic web. chapter SomeRDFS in the semantic web, pages 158–181. Springer-Verlag, Berlin, Heidelberg.
- Arruda, T., Souza, D., and Salgado, A. C. (2010). PsemRef: personalized query reformulation based on user preferences. In *Proceedings of the 12th International Conference on Information Integration and Web-based Applications & Services, iiWAS '10*, pages 683–686, New York, NY, USA. ACM.
- Athanasios, N., Christophides, V., and Kotzinos, D. (2004). Generating on the fly queries for the semantic web: The ICS-FORTH graphical RQL interface (GRQL). In *In Proc. of ISWC 2004*, pages 486–501.
- Beneventano, D., Bergamaschi, S., Guerra, F., and Vincini, M. (2007). The SEWASIE Network of Mediator Agents for Semantic Search. *J. UCS*, 13(12):1936–1969.
- Blanco, R., Ahmed, N., Hadaller, D., Sung, L. G. A., Li, H., and Soliman, M. A. (2006). A Survey of Data Management in Peer-to-Peer Systems. Technical report, David R. Cheriton School of Computer Science, University of Waterloo, Canada.
- Castano, S., Ferrara, A., Montanelli, S., and Zucchelli, D. (2003). HELIOS: a General Framework for Ontology-based Knowledge Sharing and Evolution in P2P Systems. In *Proceedings of the 14th International Workshop on Database and Expert Systems Applications, DEXA '03*, pages 597–603, Washington, DC, USA. IEEE Computer Society.
- Cybis, W. d. A., Dyck, A. F., de Oliveira, R., Pagliuso, P. B. B., Tambascia, C. d. A., Villas-Boas, A. L. d. C., and Freitas, M. (2003). Approach for the development of systematic and productive verifying lists of usability. In *Proceedings of the Latin American conference on Human-computer interaction, CLIHC 03*, pages 29–40, New York, NY, USA. ACM.
- Dix, A., Finlay, J. E., Abowd, G. D., and Beale, R. (2003). *Human-Computer Interaction (3rd Edition)*. Prentice Hall, 3 edition.
- Eclipse (2014). Eclipse - Integrated Development Environments. <http://www.eclipse.com/>.
- Franconi, E., Guagliardo, P., and Trevisan, M. (2010). An intelligent query interface based on ontology navigation. In *Proc. of the Workshop on Visual Interfaces to the Social and Semantic Web (VISSW 2010)*, Hong Kong, CN.
- Freenet (2014). The Free Network Project. <http://freenetproject.org/>.
- Halevy, A., Rajaraman, A., and Ordille, J. (2006). Data integration: the teenage years. In *Proceedings of the 32nd international conference on Very large data bases, VLDB '06*, pages 9–16, Seoul, Korea. VLDB Endowment.
- Herschel, S. and Heese, R. (2005). Humboldt Discoverer: A semantic P2P index for PDMS. In *Proceedings of the International Workshop Data Integration and the Semantic Web*, Porto, Portugal.

- Hoang, H. H. and Tjoa, A. M. (2006). The state of the art of ontology-based query systems: A comparison of existing approaches. In *In Proceedings of International Conference on Computing & Informatics, ICOCI'06*, Kuala Lumpur, MALAYSIA.
- Hyvonen, E. and Viljanen, K. (2003). Ontogator: combining view- and ontology-based search with semantic browsing. In *In Proceedings of XML*, Finland.
- Java (2014). Java language. <http://www.java.com/>.
- Jeffries, R., Miller, J. R., Wharton, C., and Uyeda, K. (1991). User interface evaluation in the real world: a comparison of four techniques. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Reaching through technology*, CHI '91, pages 119–124, New York, NY, USA. ACM.
- LabUtil (2014). Checklist ErgoList. <http://www.labiutil.inf.ufsc.br/ergolist/check.htm>.
- Li, W.-S., Candan, K. S., Hirata, K., and Hara, Y. (1997). IFQ: A Visual Query Interface and Query Generator for Object-based Media Retrieval. In *Proceedings of the 1997 International Conference on Multimedia Computing and Systems*, pages 353–, Washington, DC, USA. IEEE Computer Society.
- Makela, E., Hyvonen, E., Saarela, S., and Viljanen, K. (2004). OntoViews – A Tool for Creating Semantic Web Portals. In *In Proceedings of Third International Semantic Web Conference (ISWC2004)*, pages 797–811, Hiroshima, Japan. Springer Verlag.
- Mandreoli, F., Martoglia, R., Penzo, W., Sassatelli, S., and Villani, G. (2007). SUNRISE: Exploring PDMS Networks with Semantic Routing Indexes. In *In Proceedings of the 4th European Semantic Web Conference (ESWC)*, Innsbruck, Austria.
- Mandreoli, F., Martoglia, R., Sassatelli, S., Villani, G., and Penzo, W. (2008). Building a PDMS infrastructure for XML data sharing with SUNRISE. In *Proceedings of the 2008 EDBT workshop on Database technologies for handling XML information on the web*, DataX '08, pages 51–59, New York, NY, USA. ACM.
- Montanelli, S. and Castano, S. (2008). Semantically routing queries in peer-based systems: The h-link approach. volume 23, pages 51–72, New York, NY, USA. Cambridge University Press.
- Pereira, T. P. A. (2011). Explorando o conhecimento disponível na web semântica e o feedback do usuário para melhorar a acurácia de alinhamentos gerados em processos de ontology matching. Master's thesis, Informatics Center, Federal University of Pernambuco (UFPE/CIn), Recife, Brazil.
- Pereira, T. P. A., Pires, C. E., and Salgado, A. C. (2011). Exploring web semantic knowledge and user feedback to improve ontology matching. pages 234–238, Los Alamitos, CA, USA. IEEE Computer Society.
- Pires, C. (2009). *Ontology-based Clustering in a Peer Data Management System*. PhD thesis, Informatics Center, Federal University of Pernambuco (UFPE/CIn), Recife, Brazil.
- Prates, R. O. and Barbosa, S. D. J. (2003). Avaliação de interfaces de usuário - conceitos e métodos. volume 2, pages 245–293, Campinas, SP, BR.

- Prud'hommeaux, E. and Seaborne, A. (2008). Sparql query language for rdf. W3c recommendation, W3C. <http://www.w3.org/TR/2008/REC-rdf-sparql-query-20080115/>.
- Rogers, Y., Sharp, H., and Preece, J. (2011). *Interaction Design: Beyond Human - Computer Interaction*. Wiley, 3 edition.
- Scapin, D. L. (1990). Organizing human factors knowledge for the evaluation and design of interfaces. *Int. J. Human Computer Interaction*, 2(3):203–229.
- Souza, D. (2009). *Using Semantics to Enhance Query Reformulation in Dynamic Distributed Environments*. PhD thesis, Informatics Center, Federal University of Pernambuco (UFPE/CIIn), Recife, Brazil.
- Souza, D., Arruda, T., Salgado, A. C., Tedesco, P., and Kedad, Z. (2009). Using semantics to enhance query reformulation in dynamic environments. volume 13, pages 78–92, Riga, Latvia.
- Stojanovic, N., Studer, R., and Stojanovic, L. (2004). An approach for step-by-step query refinement in the ontology-based information retrieval. pages 36–43, Los Alamitos, CA, USA. IEEE Computer Society.
- Yang, B. and Garcia-Molina, H. (2003). Designing a super-peer network. In *In Proceedings of the 19th International Conference on Data Engineering (ICDE)*, pages 49–73, Bangalore, India.