

A Visual Query Interface for Ontology-based Peer Data Management Systems

Andreza Leite de Alencar^{1,2}, Ana Carolina Salgado¹

¹Informatic Center, Federal University of Pernambuco, Recife, Brazil

²Computer Engineering, Federal University of Vale do São Francisco, Juazeiro - BA, Brazil

andreza.leite@univasf.edu.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 System (PDMS). Many of these PDMS are semantic based and composed of data peers which exported schemas 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, which provides a simple and straightforward interaction with this type of system. It aims not only providing a natural visual query interface but also supporting precise direct manipulation of the data schemas for query generation.*

1. INTRODUCTION

User interaction for the formulation and execution of queries has been investigated for distributed and dynamic environments, such as Peer Data Management System (PDMS) [Blanco et al. 2006]. These environments are highly dynamic in an infrastructure of peers which are distributed, heterogeneous and autonomous. Many of these semantic based PDMS are composed of data peers which exported schemas are represented by ontologies [Hoang and Tjoa 2006]. There are some proposed PDMS query interfaces, but none of them addresses, in a general way, the needs of a PDMS for 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]. In this context, we propose a visual interface for PDMS which provides to users a simple and straightforward interaction through a set of features such as schema manipulation, query formulation, results visualization and network topology visualization. This features address the essential needs to interact with a PDMS.

This interface offers the concept ontologies visualization as a graph or as a hierarchy, providing a visual query formulation by ontology manipulation, selecting the concepts and combining them with constructors available in the interface. Another option to submit queries is using SPARQL Protocol and RDF Query Language (SPARQL)[Prud'hommeaux and Seaborne 2008]. SPARQL templates are available through interface and the users just need to insert the concepts in order to be queried. Also, we offer the possibility to improve the results by enriching the submitted query, using variables that represent relationships between the concepts of the query (approximation, subconcept, superconcept and aggregation). In addition to queries features, the

interface allows an organized view of the results with information about the correspondences that generated them, and the graphical visualization of the network topology identifying the peers that participated on the generated results.

The paper is organized as follows: section 2 presents the main PDMS concepts and describe the 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 experiments and evaluation of the interface; section 5 presents some related works and compare this with our approach; section 6 presents the conclusions and future works.

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 the 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 to be shared with other peers. In such systems, schema matching techniques are used to establish schema mappings (i.e., correspondences between schema elements) which form the basis for query answering and peer clustering. Schema mappings are defined between pairs of semantic neighbor peers, i.e., peers that are semantically related and previously identified by a clustering process. Queries submitted at a peer are answered with data residing at that peer and with data that is reached through mappings over the semantic neighbors.

2.1. Ontology-based PDMS

The establishment of schema mappings and consequently query answering in PDMS can be improved if semantically similar peers are put together in the overlay network. In this sense, Pires [Pires 2009] proposed a semantic-based PDMS, called 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, SPEED 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 SPEED, the peers are clustered according to the same knowledge domain, and an ontology describing the domain is available to be used as background knowledge. Ontologies are also used as a uniform conceptual representation of peer schemas, and correspondences between these ontologies are set to provide an understanding of their data sources. SPEED consider that correspondences are defined between pairs of semantic neighbor peers, i.e., peers that are semantically related as identified by a clustering process.

2.2. 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 2012]. The strengths of such topologies are exploited in order to assist peer organization in the network according to

their exported schemas. SPEED's main goal is to cluster semantically similar peers in order to facilitate the establishment of semantic mappings between peers (i.e., correspondences between schema elements) which form the basis for query answering and peer clustering [Pereira et al. 2011] and, consequently, improve query processing on a large number of data sources [Pires 2009]. Next, we present an overview of the SPEED's architecture and query processing.

The SPEED's Architecture

A DHT network is used to link particular peers that represent a certain knowledge domain. Peers are grouped according to their knowledge domain (e.g., Education and Health), forming semantic communities. Inside a 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 neighbors in the overlay network, i.e., to other semantically similar clusters. In other words, semantically similar peers are clustered 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).

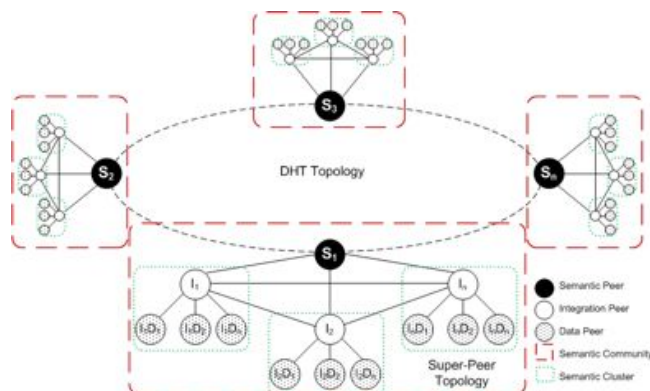


Figure 1. SPEED's Architecture [Pires 2009]

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 where it was posed. Semantic peers do not participate in query processing. Consequently, if a query is submitted at a particular semantic community the query is not forwarded to other communities. During the navigation, a query is reformulated according to previously established semantic mappings. 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].

3. THE VISUAL QUERY INTERFACE

In this work we propose a visual query interface, called VisualSPEED, for ontology-based PDMS. It aims at providing a natural visual query interface but 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 the main characteristics of it.

VisualSPEED Layers

The interface features, were defined based on SPEED system. In this context, we defined two new layers to the SPEED's architecture: (1) a user interaction layer and (2) a management layer. Figure 2 illustrates the proposed new layers and their modules on solid line and the third layer, which represents the SPEED's core, on dotted line.

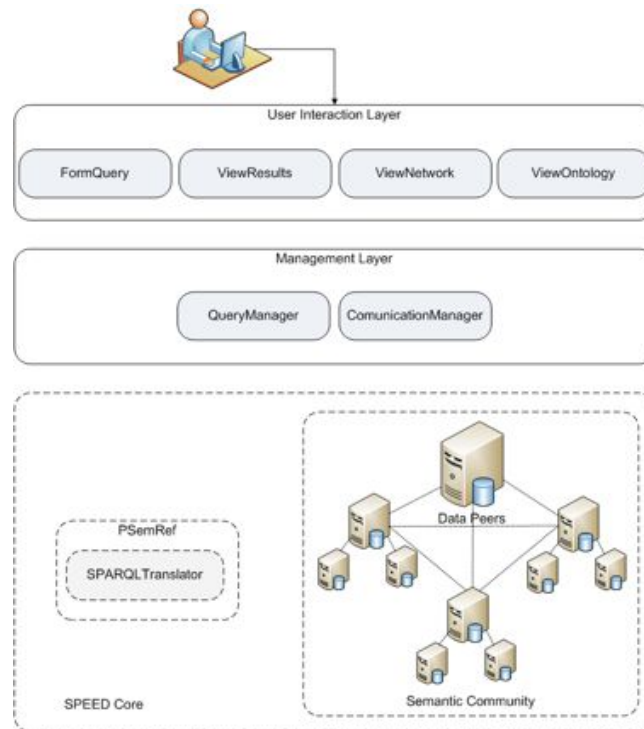


Figure 2. The architecture with interaction and manager layer - Enhancing SPEED platform

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 SPEED's network, based on direct visualization and manipulation of the local data schema, (2) for users not connected to SPEED's 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 semantic correspondences that generated them and the submission peer. These information are showed in a graphical visualization of SPEED's network topology.
- ViewNetwork - responsible for displaying the network topology enabling the user to know the neighbors of the peer which he is interacting.

The Management Layer is composed by two modules responsible for the communication between the User Interaction and the SPEED's core layers:

- QueryManager - responsible for processing 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 SPEED core. This module acts as a proxy between the user interaction modules and other SPEED's components required for interaction features.

The SPEED's core contains relevant components that are not in the scope of this work:

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

Analysing the architecture, it is possible 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 ultimately comprehensible to the formulation of queries 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 for specification. For the prototype we have considered two different types of users: the participant user in SPEED system (PU, for short) and the non-participant user in the system (NPU).

Implementation and Usage Issues

The interface was developed and validated in the environment of the PDMS SPEED [Pires 2009]. All modules were implemented in Java programming language [Oracle 2012], along with Integrated Development Environment (IDE) Eclipse [Eclipse 2012].

The system will provide for both users four items in the menu: Home, Query, Network and Documentation where Home and Documentation are default for all 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 type of user. A user PU



Figure 3. The query interface with a graph visualization of the ontology

accesses the system via SPEED data peer or integration peer. These users are able to visualize their data schema in the form of graph or as a hierarchy of terms. Figure 3 shows the query interface with a preview of ontology-shaped graph. This graphical visualization allows user to manipulate the ontology emphasizing points of interest by navigating the graphical structure or setting levels of depth desired.

The query formulation is done through the selection of concepts in the graphical representation of the ontology. The selected concepts will be showed on query composition field, which is in the upper right side of Figure 3. 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 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, subconcept, superconcept and aggregation)[Arruda et al. 2010, Souza et al. 2009]. These variables represent semantic relationships between the concepts of the query represented by the generated semantic correspondences. The enrichment of the query is shown in the bottom right side of Figure 3.

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 concepts 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 a data peer in the SPEED network, he must first connect to a data peer to have access to his data schema

and proceed with the composition 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 chosen domain, and the user must choose a starting peer for submission of queries. After selecting 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, described above.

After query execution, the system displays the results organized in a table shown in the bottom right side of Figure 3. This result visualization allows users to see detailed information about them, such as semantic correspondences, used by the reformulation process through data peers. Users can also see the identification of data peers that originated the result presented as a graphical representation of the SPEED network topology. The visualization of the network topology is also presented through the Network menu.

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. Providing the users with the possibility of removing correspondences is a good way to obtain knowledge about incorrect correspondences [Pereira 2011].

4. Experiments and Results

Regarding to the assessment of the interface and verification of compliance with usability criteria, experiments were performed with two types of users (experts and non-experts). In this section we describe the scenario used in the experimentation of the interface. The domain used for user interaction during the experiments was *Education*. For these experiments, we perform a query $Q_1 = Student \sqcup Project$, which seeks projects or students, by selecting the concepts in the graphic visualization of the ontology and the constructor available on the interface. This same query was submitted in two ways: (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 , using the same concepts but in SPARQL option. For this query, we used the SPARQL template *Union*.

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 where the query was executed. Is important to point that the interface usage, described in section 3, improves query formulating and results visualization as the evaluation will show in next section.

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 it is used. In addition, functionality tests are needed to verify the robustness of the implementation. Evaluation of usability of an 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 that evaluated the usability, assuring that the interface is, in general, in ac-

cordance with expected usability criteria. Another one that evaluated the functionalities available in the interface. In this section we will describe the 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 perform the experiments using the interface and evaluating their features. These users are divided into two groups: (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 provide to users a script with a step-by-step of the experiments to be performed. These experiments made use of the queries Q_1 and Q_2 , described above, being submitted in three ways: (1) restricted mode, only with inclusion of concepts 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.

To get the evaluation of the users 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 below.

1. Domain and peer selection.
2. A sign up form for a new peer entering the SPEED system.
3. Display of ontology as a graph.
4. Display of ontology as a concept hierarchy.
5. Visualization of SPEED's network topology.
6. Composition of queries and the 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 4 and 5 display the results of these experiments by expert users and non-expert users, respectively. The graphics reports the percentage of responses (Poor, Good and Excellent) for each question about system's functionality.

Looking at the graphic, we observe that the majority of users, experts and non-experts, answered the 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, confirming that the user interface provides a simple and straightforward interaction with the system.

Few users (non-experts) rated negative features such as: submission of SPARQL query, use of enrichment variables and visualization of semantic information of query results. But 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.

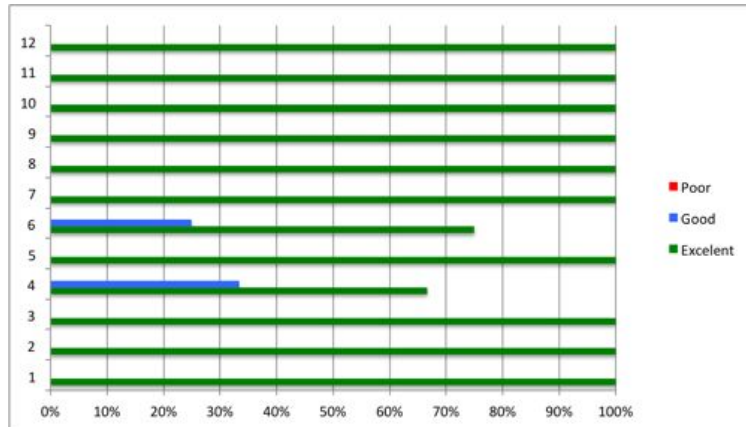


Figure 4. Results of experts' evaluation.

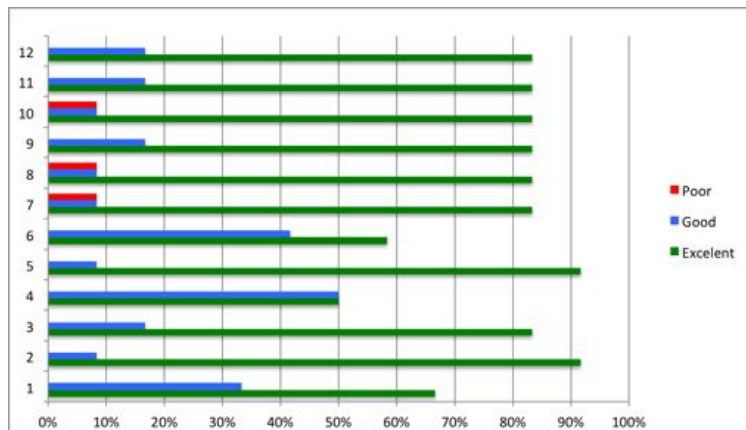


Figure 5. Results of non-experts' evaluation.

5. RELATED WORKS

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

created by projection of ontologies 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 ontologies based query interfaces, we have found an approach called SUNRISE [Mandreoli et al. 2007], which allows the user to explore the most promising way to route queries in the network.

Among the approaches considered, all of them include the formulation and submission of queries but do not offers these features as a visual interface. Table 1 presents a comparison between the approaches analysed and our proposal with some more characteristics like: Query Formulating (proposed techniques for formulating and submitting queries); Results Visualization (proposed techniques to results visualization); Network Visualization (proposed visualization of the network topology); and User (enabled interaction of various user types). 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 the VisualSPEED does.

Table 1. Contribution table of our interface in comparison to the existing approaches

Approaches	Query Formulating	Results Visualization	Network Visualization	Codification	Types of User
Query Tool	X	-	-	-	X
Graphical RQL	X	-	-	-	-
SEWASIE	X	-	-	X	-
Ontogator	X	X	-	X	-
OntoViews	X	-	-	X	-
SUNRISE	X	-	X	X	-
VisualSPEED	X	X	X	X	X

6. CONCLUSIONS AND FUTURE WORKS

Analysing the results of the evaluations, we can conclude that our approach is a visual interface that provides the functionality and transparency necessary for user interaction with a PDMS. We have contributed in some aspects such as: defining a PDMS interface that enables the interaction of various types of user; providing the formulation of queries in a visual way by manipulating the graphical visualization of ontologies; the usability of constructors, SPARQL instructions and enrichment variables to query composition; the visualization of the results in an organized way allowing access to detailed information about them; the possibility to assess the semantic correspondences that was used to generate the results of the queries; the identification of peers that have participated in the results in the network topology; the visualization of the network topology with all the peers and semantic communities that compose the system; the implementation, evaluation and experimentation performed.

The interface specification was developed to satisfy any PDMS needs, but the implemented prototype is strongly coupled with SPEED's environment and requires several code modifications to run in another PDMS environments. Then, we want to develop a multi PDMS interface in which the required modifications will be just configuration files or simple code extensions.

In addition, the approach deserves some extensions and improvements, such as 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., and Guerra, F. (2007). The sewasie network of mediator agents for semantic search. *Computer*, 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.
- 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–, Washington, DC, USA. IEEE Computer Society.
- Dix, A., Finlay, J. E., Abowd, G. D., and Beale, R. (2003). *Human-Computer Interaction (3rd Edition)*. Prentice Hall, 3 edition.
- Eclipse (2012). Eclipse - integrated development environments. Disponível em: <http://www.eclipse.com/>.
- Franconi, E., Guagliardo, P., and Trevisan, M. (2010). An intelligent query interface based on ontology navigation. *proceedings from Workshop on Visual Interfaces to the Social and Semantic Web (VISSW2010)*.
- Freenet (2012). The free network project.
- 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. 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*.
- 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 Proc. of ICOC106*.
- Hyvonen, E. and Viljanen, K. (2003). Ontogator: combining view- and ontology-based search with semantic browsing. In *In Proceedings of XML*.

- 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 THE THIRD INTERNATIONAL SEMANTIC WEB CONFERENCE (ISWC2004)*, pages 797–811. Springer Verlag.
- Mandreoli, F., Martoglia, R., Penzo, W., Sassatelli, S., and Villani, G. (2007). Sunrise: Exploring pdms networks with semantic routing indexes.
- 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. *Knowl. Eng. Rev.*, 23:51–72.
- Oracle (2012). Java language. Disponível em: <http://www.java.com/>.
- 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, Federal University of Pernambuco (UFPE/CIn).
- Pereira, T. P. A., Pires, C. E., and Salgado, A. C. (2011). Exploring web semantic knowledge and user feedback to improve ontology matching. *Database and Expert Systems Applications, International Workshop on*, 0:234–238.
- Pires, C. (2009). *Ontology-based Clustering in a Peer Data Management System*. PhD thesis, Tese de Doutorado, Centro de Informática - UFPE.
- Prates, R. O. and Barbosa, S. D. J. (2003). Avaliação de interfaces de usuário - conceitos e métodos. *Jornada de Atualização em Informática do Congresso da Sociedade Brasileira de Computação. Campinas: SBC*, v. 2:p. 245–293.
- 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.
- Souza, D., Arruda, T., Salgado, A. C., Tedesco, P., and Kedad, Z. (2009). Using semantics to enhance query reformulation in dynamic environments. *13th East European Conference on Advances in Databases and Information Systems (ADBIS'09), Riga, Latvia.*, 13:pp. 78–92.
- Stojanovic, N., Studer, R., and Stojanovic, L. (2004). An approach for step-by-step query refinement in the ontology-based information retrieval. *Web Intelligence, IEEE / WIC / ACM International Conference on*, 0:36–43.
- Yang, B. and Garcia-Molina, H. (2003). Designing a super-peer network. In *ICDE*, pages 49–.