

An Ontology-Based Framework for Geographic Data Integration

Vânia M.P. Vidal¹, Eveline R. Sacramento¹, José Antonio Fernandes de Macêdo^{1,2}
Marco Antonio Casanova³

¹Universidade Federal do Ceará, Department of Computing, Brazil
{eveline, vvidal, jose.macedo}@lia.ufc.br

²EPFL - Ecole Polytechnique Fédérale, Database Laboratory, Switzerland
jose.macedo@epfl.ch

³Department of Informatics – Pontifical Catholic University of Rio de Janeiro
Rio de Janeiro, RJ – Brazil
casanova@inf.puc-rio.br

Abstract. Ontologies have been extensively used to model domain-specific knowledge. Recent research has applied ontologies to enhance the discovery and retrieval of geographic data in Spatial Data Infrastructures (SDIs). However, in those approaches it is assumed that all the data required for answering a query can be obtained from a single data source. In this work, we propose an ontology-based framework for the integration of geographic data. In our approach, a query posed on a domain ontology is rewritten into sub-queries submitted over multiples data sources, and the query result is obtained by the proper combination of data resulting from these sub-queries. We illustrate how our framework allows the combination of data from different sources, thus overcoming some limitations of other ontology-based approaches. Our approach is illustrated by an example from the domain of aeronautical flights.

Keywords: Keywords: data integration, schema mappings, geographic information retrieval, query processing, Web Feature Service, ontologies.

1 Introduction

Spatial Data Infrastructures (SDIs) provide access, reuse and integration of geographic information (GI) from multiple sources. Service providers currently offer access to geospatial data and expose basic processing functionality using Web services technology [1, 2, 3, 6]. This strategy not only offers a standardized, flexible and transparent way to publish underlying data but it also hides details of data access and retrieval from the application. In OGC-compliant SDIs, geospatial data are served via Web Feature Services (WFS). Each WFS offers a feature type schema (FTS), which is the XML schema of the feature type exported by the service. Users can query and update data sources through an FTS. The specifications provided by the

Open Geospatial Consortium (OGC) enable syntactic interoperability and cataloguing of GI.

In any data sharing architecture, including SDIs, reconciling semantic heterogeneity is a key issue. No matter whether the query is issued or whether the data is shared, the semantic differences between data sources need to be reconciled. Typically, semantic mappings are used to define how translate data from one data source into another, preserving the semantics of the data or, alternatively, to rewrite a query posed on one source into a query on another source. However, the specification of these mappings is labor intensive and error prone, representing over half of the effort spent in a typical data integration scenario. Moreover, the problem of semantic heterogeneity is exacerbated when dealing with semi-structured data due to its flexibility in adding new attributes and, consequently, generating more schema variations. One possible approach to overcome this problem is the explicitation of knowledge by means of ontologies [5]. In this sense, the idea is to use ontologies to describe terms of the domain and the data WFSs services.

Current research [2,3] in the geospatial context have proposed the use of DL ontologies for enhancing discovery and retrieval of geographic information. The framework proposed in [3] adopts a hybrid ontology approach [5], where each feature type schema offered via WFS is described by specific application concepts that are built using properties and classes from a shared vocabulary. The shared vocabulary is represented by a domain ontology that contains basic terms (the primitives) of a domain. These terms are combined to describe the semantic of feature types in separate application ontologies. It is assumed that all actors within a domain share a common understanding of the concepts contained in the domain ontology. In this framework, the requester formulates a query using terms from the domain ontology. Reasoning services are used to determine whether existing application concepts (describing feature types) are a match for the query concepts. When an appropriate feature type is discovered, the query can be used to generate a request to retrieve data from its WFS. The translation of the query into the actual WFS query, which is formulated in terms of the feature type schema, is based on so-called registration mappings [4], which map the structure of the schema to ontology concepts. The restriction of this approach is that a data source is discovered only if it contains all the information required for answering the user's question.

In [4], it is proposed a methodology that uses rules for both the discovery of data sources and, based on the discovered data, answering queries in SDIs. Their approach allows inferences that use relationships between individuals and the combination of data from different sources. Query answering is realized in three steps: First, schema mapping and domain rules are used in the discovery of appropriate data sources that can answer a specific user query. Then, the knowledge base has to be populated with data of the relevant data sources. Finally, using domain rules, new knowledge is inferred to answer the user query. The major drawback of this approach is that a large amount of data that is materialized in the knowledge base may not be relevant to the user query.

In this paper, we propose a framework that deals with the situation where data from several data sources have to be combined in order to answer a given question. In our approach, a query formulated in terms of a domain ontology is rewritten into sub-

queries submitted over multiples data sources, and the query results are obtained by the proper combination of data resulting from these sub-queries.

The remainder of the paper is structured as follows. Section 2 describes our framework for integration of geographic data. Section 3 describes the proposed approach with the help of an example. Section 4 presents the conclusions and directions for future research.

2 A Framework for Geographic Data Integration

Figure 1 describes the main components of the proposed framework. The mediated schema is represented by a domain ontology (DO), which provides a conceptual representation of the application domain (a global shared vocabulary). Each feature type schema, offered via a WFS, is described by an application ontology (AO) whose vocabulary is restricted to be a subset of the vocabulary of DO. The Global Ontology consists of the union of the application ontologies, and a set of axioms that define inter-ontology properties. The *mediated mapping* defines the concepts and properties of the domain ontology in terms of the vocabularies of the global ontology, whereas the *local mappings* define the classes and properties of the application ontologies in terms of the elements of its feature type schemas.

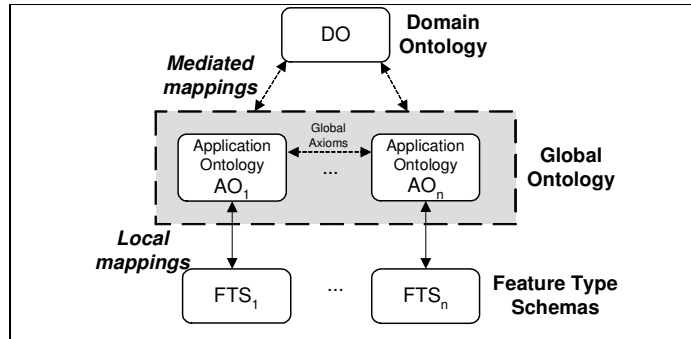


Fig. 1. Ontology-based Architecture for discovery and retrieval of geographic information

In our approach, the global ontology plays a key role in order to deal with data integration. Application ontologies help breaking the query answering problem into two sub-problems, as discussed in Section 4. They are also a notational convenience to divide the definition of the mappings into two stages: the definition of the mediated mapping and the definition of the local mappings.

In order to represent ontologies and mappings, we adopt a family of logics called Description Logics (DL) [7,8]. The following definition formally introduces the notion of mediated environment.

Definition 2.1: (Mediated Environment) A mediated environment is a 6-tuple $ME = (DO, FTS_k, AO_k, \gamma_k, AO_k, \gamma)$, $k=1, \dots, n$, where

- DO is a *domain ontology*, which represents the mediated schema. We assume that the classes and properties in DO are C_1, \dots, C_u and P_1, \dots, P_v .
- for each $k=1, \dots, n$,
 - FTS_k is a feature type schema
 - AO_k is an *application ontology*, which describes exactly the feature type FTS_k. The vocabulary of AO_k is a subset of the vocabulary of DO. We adopt namespace prefixes to distinguish the occurrence of a symbol in the DO vocabulary from the occurrence of the same symbol in the vocabulary of AO_k. We assume that:
 - the classes and properties in DO are C_1, \dots, C_u and P_1, \dots, P_v . So, for each class C_i (or property P_j) in the vocabulary of DO, we denote the occurrence of C_i (or P_j) in the vocabulary of AO_k by AO_k: C_i (or AO_k: P_j)
 - (*Domain Disjointness Assumption*) for any interpretation ξ_i and ξ_j for the alphabets of AO_i, AO_j, ξ_i and ξ_j have disjoint domains, for each $i, j \in [1, k]$, with $i \neq j$
 - γ_k is a set of correspondence assertions, called a *local mapping*, each one of the form $A \equiv T_k / \delta$, where A is a class or property of AO_k, T_k is the feature type schema described by AO_k, and δ is a path of T_k
- GO is the *global ontology*, which consists of the union of the application ontologies AO_k, $k=1, \dots, n$, and a new set of *inter-ontology* properties, introduced by definition.
- γ is the *mediated mapping*, which defines (some of) the γ defines the classes and properties of DO in terms of the classes and properties of the GO, and is such that:
 1. for each $i=1, \dots, u$, the mapping γ contains a definition of the form

$$C_i \equiv c_1 \sqcup \dots \sqcup c_n \tag{1}$$

where c_k is a class of GO, $k=1, \dots, m$.

2. for each $j=1, \dots, v$, the mapping γ contains a definition of the form

$$P_j \equiv p_1 \sqcup \dots \sqcup p_m \tag{2}$$

where p_k is a property of GO, $k=1, \dots, m$.

In following, we explain in more detail our mediated environment through a data integration example, adapted from [4].

Feature type Schemas

In this example, we assume that the user provides feature type schemas. We consider three data sources. The first data source is based on the Digital Aeronautical Flight Information File (DAFIF), the second is based on the Aeronautical Information Exchange Model (AIXM) and the third data source concerns Aircraft Database (AIRFRAMES). The DAFIF and AIXM data sources provide information about airports and their runways. Particularly, the DAFIF data source has airports with runway length less than 5,000 meters. The AIRFRAMES data source provides information about aircrafts. Figure 2 shows the feature type schemas exported by these data sources via Web Feature Services.

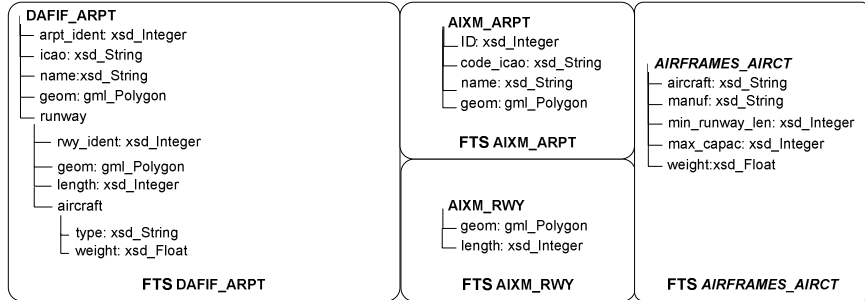


Fig.2. Feature Type Schemas.

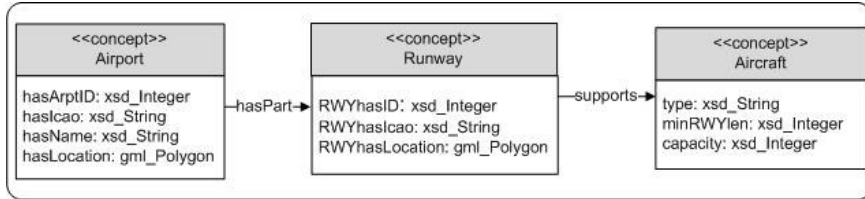


Fig. 3. Domain Ontology AirportOnto.

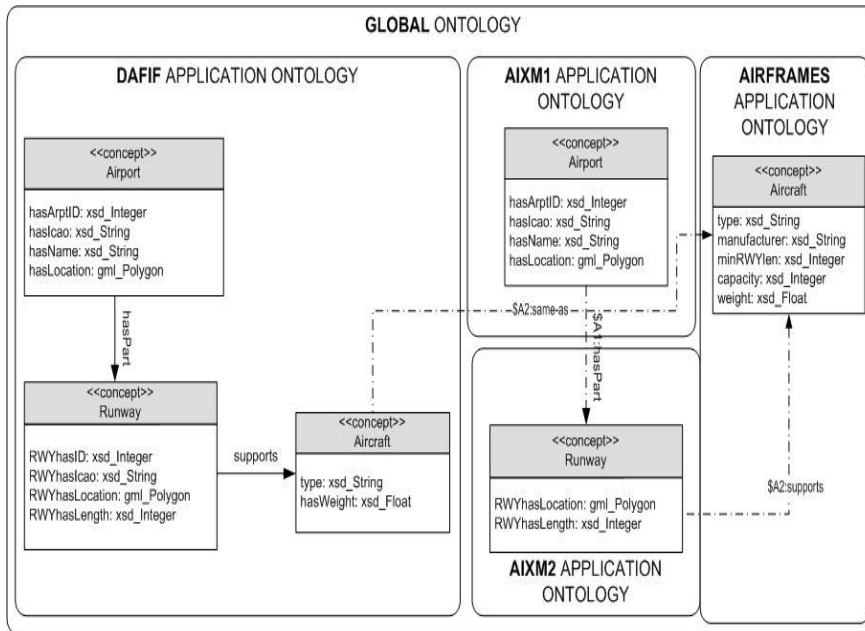


Fig.4. Application Ontologies and Global Ontology

Domain, Global, and Application Ontologies

In our approach, we assume that the user provides the domain ontology, and that there is an application ontology described with the shared vocabulary of the domain ontology, for each feature type schema, offered via WFS.

Figure 3 shows the domain ontology AirportOnto, which provides a suitable vocabulary covering the main concept of our restricted aeronautical flight domain. Figure 4 shows the global ontology, which contains the union of application ontologies for the FTSs in Figure 2, and the *inter-ontology properties* defined in Figure 5. The property \$A1:hasPart is defined as the combination (join) of AIXM1:Airport and AIXM2:RunWay using a topological binary relation (*inside*) on the geometry properties (see line 1 of Figure 5). Likewise, the property \$A2:supports is obtained by the combination of AIXM2:RunWay and AIRFRAMES:Aircraft using the binary relation *greater_than* on the lengths properties (see line2 of Figure 5). The axiom in line 4 defines that the property \$A4:capacity is obtained by the composition of DAFIF:type, AIRFRAMES:type and AIRFRAMES:capacity.

Mediated and Local Mappings

Figure 6 shows the mediated mappings, and Figure 7 shows the local mappings defining the classes and properties of the DAFIF application ontology in terms of its FTS. Due to space limitation, the local mappings for the other application ontologies are omitted here.

As already mentioned, the DAFIF data source has airports with runway length less than 5,000 meters. Translating this constraint to the vocabulary of the DAFIF application ontology, we have the following constraint (expressed in DL):

$$\text{DAFIF:Airport} \sqsubseteq \exists \text{DAFIF:hasPart} . (\exists \text{DAFIF:hasLength} . \{<5000\})$$

3 Query Processing

We propose a two-step strategy for answering a query Q posed on the domain ontology, summarized as follows:

1. The user's query is decomposed into a set of elementary sub-queries expressed in terms of the application ontologies. Each such (elementary) sub-query aims at extracting data from a single application ontology. The result of this step is a query expressed as unions and joins over elementary sub-queries. This step is performed using the mediated mappings.
2. Sub-queries resulting from the previous step are rewritten in terms of FTSs with the help of the local mappings. Hence, we obtain a *global execution plan*, which is a combination of WFS queries using joins, unions and other (possibly spatial) operations.

1. $\$A1:hasPart \equiv AIXM1:hasLocation \circ \textit{inside} \circ AIXM2:RWYhasLocation^-$
2. $\$A2:supports \equiv AIXM2:RWYhasLength \circ \textit{greater_than} \circ AIRFRAMES:minRWYLength^-$
3. $\$A3:same-as \equiv DAFIF:type \circ AIRFRAMES:type^-$
4. $\$A4:capacity \equiv \$A3:same-as \circ AIRFRAMES:capacity$

Fig. 5. Inter-Ontology Properties.

Concept Mappings:

1. Airport \equiv DAFIF:Airport \sqcup AIXM1:Airport
2. Runway \equiv DAFIF:Runway \sqcup AIXM2:Runway
3. Aircraft \equiv DAFIF:Aircraft \sqcup AIRFRAMES:Aircraf

Property Mappings:

4. hasLocation \equiv DAFIF:hasLocation \sqcup AIXM1:hasLocation
5. hasPart \equiv DAFIF:hasPart \sqcup \$A1:hasPart
6. hasLength \equiv DAFIF:RWYhasLength \sqcup AIXM2:RWYhasLength
7. supports \equiv DAFIF:supports \sqcup \$A2:supports
8. capacity \equiv \$A4:capacity \sqcup AIRFRAMES:capacity

Fig. 6. Mediated Mappings.

Concept Mappings:

DAFIF:Airport \equiv DAFIF_ARPT

DAFIF:Runway \equiv DAFIF_ARPT/runway

DAFIF:Aircraft \equiv DAFIF_ARPT/runway/aircraft

Property Mappings:

DAFIF:hasArptID \equiv DAFIF_ARPT / arpt_ident

DAFIF:hasIcao \equiv DAFIF_ARPT / icao

DAFIF:hasName \equiv DAFIF_ARPT / name

DAFIF:hasLocation \equiv DAFIF_ARPT / geom

DAFIF:hasPart \equiv DAFIF_ARPT / runway

DAFIF:RWYhasId \equiv DAFIF_ARPT / runway / rwy_ident

DAFIF:RWYhasIcao \equiv DAFIF_ARPT / runway / icao

DAFIF:RWYhasLocation \equiv DAFIF_ARPT / runway / geom

DAFIF:RWYhasLength \equiv DAFIF_ARPT / runway / length

DAFIF:supports \equiv DAFIF_ARPT / runway / aircraft

DAFIF:type \equiv DAFIF_ARPT / runway / aircraft / type

DAFIF:hasWeight \equiv DAFIF_ARPT / runway / aircraft / weight

Fig. 7. Local Mappings from the feature type schema DAFIF_ARPT to its Application Ontology.

In our approach, a query has the form: $Q \equiv A \sqcap e$, where A represents an atomic concept and e represents a restriction over A . For example, consider a query Q that selects airports that have a runway whose length is greater than 13,000 meters, and support aircrafts with capacity greater than 300 passengers. This query in the DL *ALCQI* syntax is shown below:

$$Q \equiv \text{Airport} \sqcap e$$

$$\text{where } e = \exists \text{hasPart} . (\exists \text{hasLength} . \{>13000\} \sqcap \exists \text{supports} . (\exists \text{capacity} . \{>300\}))$$

Figure 8 illustrates the steps necessary for decomposing the query Q in sub-queries expressed in terms of the application ontologies.

<p>1. Rewrite the query Q using the mediated mappings:</p> $Q' \equiv (\text{DAFIF:Airport} \sqcup \text{AIXM1:Airport}) \sqcap e^1$ <p>where $e^1 = \exists (\text{DAFIF:hasPart} \sqcup \text{\\$A1:hasPart}) .$</p> $(\exists (\text{DAFIF:hasLength} \sqcup \text{AIXM2:RWYhasLength}) . \{>13000\}) \sqcap$ $(\exists (\text{DAFIF:supports} \sqcup \text{\$A2:supports}) .$ $(\exists (\text{\$A4:capacity} \sqcup \text{AIRFRAMES:capacity} . \{>300\})))$
<p>2. Apply the Distributive Property over Union:</p> $Q' \equiv Q_1 \sqcup Q_2$ <p>where $Q_1 \equiv (\text{DAFIF:Airport} \sqcap e^1)$</p> $Q_2 \equiv (\text{AIXM1:Airport} \sqcap e^1)$
<p>3. Simplify e^1 by using domain disjointness axiom:</p> $Q_1 \equiv \text{DAFIF:Airport} \sqcap e_1^1$ <p>where $e_1^1 = \exists \text{DAFIF:hasPart} . (\exists \text{DAFIF:hasLength} . \{>13000\}) \sqcap$</p> $\exists (\text{DAFIF:supports} . (\exists \text{\$A4:capacity} . \{>300\}))$ $Q_2 \equiv \text{AIXM1:Airport} \sqcap e_2^1$ <p>where $e_2^1 = \exists \text{\\$A1:hasPart} . (\exists \text{AIXM2:RWYhasLength} . \{>13000\}) \sqcap$</p> $\exists (\text{\$A2:supports} . (\exists \text{AIRFRAMES:capacity} . \{>300\}))$
<p>4. Replace the inter-ontology properties in e_1^1 and e_2^1:</p> $Q_1 \equiv \text{DAFIF:Airport} \sqcap e_1^2$ <p>where $e_1^2 = \exists \text{DAFIF:hasPart} . (\exists \text{DAFIF:hasLength} . \{>13000\}) \sqcap$</p> $\exists (\text{DAFIF:supports} .$ $(\exists ((\text{DAFIF:type} \circ \text{AIRFRAMES:type} \bar{\circ} \text{AIRFRAMES:capacity}) . \{>300\})))$ $Q_2 \equiv \text{AIXM1:Airport} \sqcap e_2^2$ <p>where $e_2^2 = \exists (\text{AIXM1:hasLocation} \circ \textit{inside} \circ \text{AIXM2:RWYhasLocation} \bar{\circ}) .$</p> $(\exists \text{AIXM2:RWYhasLength} . \{>13000\}) \sqcap$ $\exists (\text{AIXM2:RWYhasLength} \circ \textit{greater_than} \circ \text{AIRFRAMES:minRWYLength} \bar{\circ}) .$ $(\exists (\text{AIRFRAMES:capacity} . \{>300\}))$

5. Apply property of role composition to decompose e_1^2 and e_2^2 into sub-queries where each sub-query is applied over a single application ontology.

$Q_1' \equiv \text{DAFIF:Airport} \sqcap e_1^3$
 where $e_1^3 = \exists \text{DAFIF:hasPart} . (\exists \text{DAFIF:hasLength} . \{>13000\} \sqcap \exists (\text{DAFIF:supports} . (\exists \text{DAFIF:type} . Q_3))$
 where $Q_3 \equiv \exists (\text{AIRFRAMES:type} \circ \text{AIRFRAMES:capacity} . \{>300\})$

$Q_2' \equiv \text{AIXM1:Airport} \sqcap e_2^3$
 Where $e_2^3 \equiv \exists (\text{AIXM1:hasLocation} \circ \textit{inside} . Q_4 \sqcap \exists ((\text{AIXM2:RWYhasLength} \circ \textit{greater_than}) . Q_5$
 where $Q_4 \equiv \exists \text{AIXM2:RWYhasLocation} \bar{\cdot} . (\exists (\text{AIXM2:RWYhasLength} . \{>13000\})$
 $Q_5 \equiv \exists \text{AIRFRAMES:minRWYLength} \bar{\cdot} . (\exists (\text{AIRFRAMES:capacity} . \{>300\}))$

6. Apply distributive property to restrictions in e_1^3 in order to partition into sub-conditions so that are possible to check its consistency (based on the application ontology constraints).

$Q_1' \equiv \text{DAFIF:Airport} \sqcap e_1^4$
 where $e_1^4 = \exists \text{DAFIF:hasPart} . (\exists \text{DAFIF:hasLength} . \{>13000\} \sqcap \exists \text{DAFIF:hasPart} . (\exists (\text{DAFIF:supports} . (\exists \text{DAFIF:type} . Q_3))$
 where $Q_3 \equiv \exists (\text{AIRFRAMES:type} \circ \text{AIRFRAMES:capacity} . \{>300\})$

7. Remove from Q' the subqueries that are not consistent (a query is not consistent if its result is empty for any database state).

By reasoning tasks, we can show that the query Q_1' is not consistent (from constraint “ $\text{DAFIF:Airport} \sqsubseteq \exists \text{DAFIF:hasPart} . (\exists \text{DAFIF:hasLength} . \{<5000\})$ ”). So, we have that:

$Q' \equiv Q_2'$

Fig. 8. Query Answering Example.

After step 7, each subquery Q_i' will be converted into a WFS query and submitted over one data source. Data resulting from all sub-queries will be combined and encoded in XML format to produce the query's result. Finally, the resulting XML data will be used to populate the domain ontology. In the literature, the translation from XML to RDF/OWL is often called “lifting”, and some solutions are already provided [9]. Due to space limitation, this research topic will be not discussed in this paper.

4 Conclusion

In this paper, we have presented an ontology-based framework for integration of geographic data. This framework takes a query on domain ontology and rewrites it into sub-queries submitted over multiples data sources. The query's result is obtained by the proper combination of data resulting from these sub-queries. We have illustrated, through an example, how our framework allows the combination of data

from different sources, thus overcoming some limitations of other ontology-based approaches.

We showed in Section 4 how to decompose a query over the domain ontology in sub-queries expressed in terms of the application ontologies using the mediated mappings. Our approach takes advantage of DL reasoning to discard sub-queries that are not consistent.

Although our present work deals with some spatial aspects (e.g. FTS, spatial locations), we are aware that this approach can be applied to other domains. As a future work, we intend to investigate how to generate application ontology and mappings. Besides, we want to implement and evaluate our query processing algorithm. In addition, we plan to study how to optimize query processing by incorporating progressive reasoning evaluation. Last but not least, we intend to investigate, in the near future, how to express spatial operations as built-in properties within the domain ontology.

References

1. Essid, M., Boucelma, O., Colonna, F., Lassoued, Y.: Query processing in a Geographic Mediation System. In: Proceedings of GIS, pp. 101--108 (2004)
2. Klien, E., Fitzner, D. I. and Maué, P.: Baseline for Registering and Annotating Geodata in a Semantic Web Service Framework. In: Proceedings of the 10th Conference on Geographic Information Science, Aalborg, Denmark (2007)
3. Lutz, M.: Ontology-based Discovery and Composition of Geographic Information Services. Phd Thesis, Institut für Geoinformatik (2005)
4. Lutz, M. and Kolas, D.: Rule-based Discovery in Spatial Data Infrastructures. In: Transactions in GIS 11(3), pp. 317--336 (2007)
5. Wache, H., Vögele, T., Visser, U., Stuckenschmidt, H., Schuster, G., Neumann, H. and Hübner, S.: Ontology-based Integration of Information - A Survey of Existing Approaches. In: Proceedings of the IJCAI-01 Workshop: Ontologies and Information Sharing, pp. 108--117 (2001)
6. Xavier, E.M.A.: Serviços Geográficos baseados em Mediadores e Padrões Abertos para Monitoramento Participativo na Amazônia, Master's Thesis, INPE, Brasil (2008)
7. Calvanese, D.; Lenzerini, M.; Nardi, D.: Description Logics for Conceptual Data Modeling. In: Chomicki, J. and Saake, G. (ed.) Logics for Databases and Information Systems. Kluwer Academic Publisher (1998)
8. Casanova, M.A.; Lauschner, T.; Paes Leme, L.A.; Breitman, K.K; Furtado, A.L.: A Strategy to Revise the Constraints of the Mediated Schema. Technical Report MCC34/09, Department of Informatics, PUC-Rio (2009)
9. Akhtar, W., Kopecky, J., Krennwallner, T., Polleres, A.: XSPARQL: Traveling between the XML and RDF worlds – and avoiding the XSLT pilgrimage, 5th European Semantic Web Conference (2008)