Integrating Ecological Data Using Linked Data Principles

Ana Maria de C. Moura[1], Fabio Porto[1], Maira Poltosi[1] , Daniele C. Palazzi[1], Régis P. Magalhães[2], Vania Vidal[2]

[1]Extreme Data Lab (DEXL Lab)
National Laboratory of Scientific Computing (LNCC)
Petrópolis – RJ – Brazil
[2]Department of Computing
Federal University of Ceará (UFC)
{anamoura, fporto, maira, dpalazzi}@lncc.br, {regispires, vvidal}@lia.ufc.br

Abstract. *This paper presents a framework to manage, treat and integrate ecological data in the context of the PELD project, currently in development in Brazil. These data, which are produced and collected from different resources, are stored in distinct relational databases and transformed later into RDF triples, using a traditional relational-RDF mapping. Taxonomical, spatial and trophic relations are explored by means of ontological properties, which make it possible to discover interesting information about existing marine species of different bays in the country, illustrated by SPARQL queries. Additionally, the endpoint thus generated allows data to be accessed on the Web of data, as linked data.*

1. Introduction

Extensive information on policies, action programs, and environmental challenges in areas such as sustainable development, climate change, environmental law, and biodiversity, has become a great concern throughout the world. Different governmental agencies[1] and commissions[2,3] have been created for the purpose of defining strategies to preserve natural environment. Among these many policies, there is a strong concern on developing systems to organize and catalogue information about the existing natural reserves, such as minerals and biological ones, which involve fauna, flora and hydro resources, enabling a more accurate control of this information.

In Brazil, a great effort is being deployed in this direction through an important national project named PELD/Brazil[4] (Brazilian Long-Term Ecological Research Program). One of its main goals is to leverage ecological knowledge, so that important data can be provided to help, reinforce government decisions, and support research related to the management of natural resources, as well as to share this information among different sectors of society. PELD project currently counts on 29 collect sites, which are distributed along different Brazilian biomes, for the purpose of consolidating the existing knowledge about their composition and learning about ecosystems functioning. Having an integrated view of these ecological data sources and making

[1]http://www.environment-agency.gov.uk/
[2]http://ec.europa.eu/environment/index_en.htm
[3]http://www.princetontwp.org/environmain.html
[4]http://ppbio.inpa.gov.br/Port/projetosassociados/peld/

them available on the Web of data as a data set [Heath, Bizer 2011], would permit other ecologist researchers throughout the world to access, as well as to reference it to other data sets, dealing with similar subjects.

A PELD site can be considered as an integration of many sub-projects concerning distinct ecological issues. Since most of these PELD sites throughout the country are not still consolidated, or are in an initial development phase, in this paper we focus on the Guanabara PELD[5]. This PELD site aims at extending knowledge about the Guanabara Bay ecosystem and providing support for managing, structuring and publishing ecological data, as well as to be a source of answers to the anthropic and climatic impacts on the bay ecosystem. Currently a database project is being developed to manage, organize and access information about Guanabara Bay ecological data. However, since Guanabara PELD is developed by a large group of biologists, responsible by distinct domains (hydrology, planktons, fishes, ecology, etc.), data are produced independently, in different formats, and according to specific methodologies. Integrating and publishing all the data produced by these groups is crucial not only to provide a homogeneous view of this data, but also to make it available for other groups working in other PELDs throughout the country. This situation offers an interesting panorama to evaluate how efficient queries and reasoning will be in the face of a query federation pattern, where data are integrated according to the Linked Data (LD) strategy.

The main contribution of this work in comparison to other existing ecological information management systems (Ecoflora[6] [Cavalcanti 2005]) is: to integrate different ecological resources and to make them available on the Web of data, using LD principles; to provide reasoning capacity, i.e., to infer new information from the stored data. By providing an ontological representation of the data model, new relationships and instances may be inferred, taking into account transitive properties and hierarchies over the model concepts, allowing researchers to discover interesting data, such as, for example, information about specie's predators in different levels of a hierarchy.

In this paper we extend the integration framework [Vidal et al. 2011] and use some techniques to create the application ontologies. Query results are extracted from PELD data sources, integrated by QEF[7] framework [Porto et al 2007], and then visualized by the user as linked data.

The remainder of this paper is structured as follows. Section 2 presents related work. Section 3 presents the framework architecture designed to integrate PELD resources. Section 4 describes some PELD application scenarios that will be used as study case for integration. Section 5 describes the scenario ontologies generated at each level of the proposed architecture, as well as the mappings rules between the domain and application ontologies. Section 6 shows how to answer user´s queries in this architecture as linked data, by presenting a query example over different PELD resources, executed in SPARQL. Finally, Section 7 concludes the paper with suggestions for future work.

---

[5] http://www.lncc.br/peldguanabara/index.php
[6] http://www.ecoflora.co.uk/
[7] http://146.134.234.248/QEF/index.html

2. Related Work

Several works such as Ecoflora[8], NRCS[8], AEZ[9], [Cavalcanti 2005], [Campos et al. 2009], [Manzi 2009] have been proposed to manage and share ecological data. However, they do not perform data integration on LD using multiple data sources. They also do not address inference provided by the use of ontologies or semantic web approaches. This paper intends to fill this gap, by proposing an integration approach based on LD, which enables ecological data to be analyzed, inferred and queried from different PELD sources. Below we present related works that address data integration over LD.

There are two possible approaches for data integration: materialized and virtual. The first approach collects, stores and accesses data in a central database. The main disadvantage of this approach is the replication of data, which in addition requires additional storage space and does not ensure the use of updated data in relation to the original datasources. LDIF [Schultz et al. 2011] is a framework that provides data integration through the use of the materialized approach. On the other hand, the virtual approach enables the execution of federated queries over a fixed set of datasources. Our work uses both the materialized and the virtual data integration approach. Jena ARQ[10] SPARQL, DARQ [Quilitz and Leser, 2008], SemWIQ [Langegger, 2010] and FedX [Schwarte 2011] are examples of systems that provide transparent access to RDF data sources, whose data can be retrieved using SPARQL. While some of them, such as SemWIQ, allows RDF schema or OWL ontologies to be used to describe the datasources, FedX transforms the original query into a federated query over the source ontologies. However, none of these tools can execute queries over a domain ontology with mappings for specific application ontologies.

The integration of scientific data in the context of linked data using the virtual approach has been discussed in [Gray et al. 2008]. In that paper the authors discuss the integration of astronomic databases using RDF as a common schema language and SPARQL as a query language. The authors adopt a peer-to-peer integration strategy, avoiding a global view agreement. In the proposed view, each database is exposed in RDF and alignment mappings define associations between databases.

The integration tools presented in this section require the manual definition of the datasources used in each query. It is also necessary to rewrite queries when a datasource schema changes. However, the generation of federated query plans from queries over a Domain Ontology can accomplish the semantic integration in virtual and automatic way. Queries over the domain ontology are also simpler and more stable than if they were made directly over the application ontologies.

3. Integration Architecture

The need to produce data in PELD projects in a homogenous format is a fundamental requirement when considering the generation of an integrated view of Brazilian ecosystems. In this context, RDF (Resource Description Framework) [Manolla, Miller 2004] has been used as a powerful strategy to interoperate, reason and publish data, besides enabling these data to connect with

[8] http://plants.usda.gov/java/

[9] http://www.fao.org/nr/land/databasesinformation-systems/aez-agro-ecological-zoning-system/en/

[10] http://jena.apache.org/documentation/query/

other resources of similar domains. Additionally, it enables the exploration and association among data, making use of SPARQL [Prud´hommeaux, Seaborne 2008].

Nevertheless, although the great benefits of RDF, there is a great concern when using it to deal with large volumes of data, since it may degrade performance [Gray et al. 2009]. This is why a current adopted strategy is to store data in relational databases. Moreover, publishing data according to the Linked Data best practices [Heath, Bizer 2011] solves part of the integration problem, which is to make data available in a common format. Ontologies come as a rescue ground from which integration becomes possible. They provide a common vocabulary to be shared among the different data sources. Thus, one needs to combine the publication of source data according to the LD best practices using RDF with a common shared vocabulary expressed as a domain ontology.

Figure 1 presents a three-level architecture used to integrate relational schemas as LD. It is based on mappings according to a mediated approach, and it has been extended from [Vidal et al 2011] to integrate the PELD databases as described below.

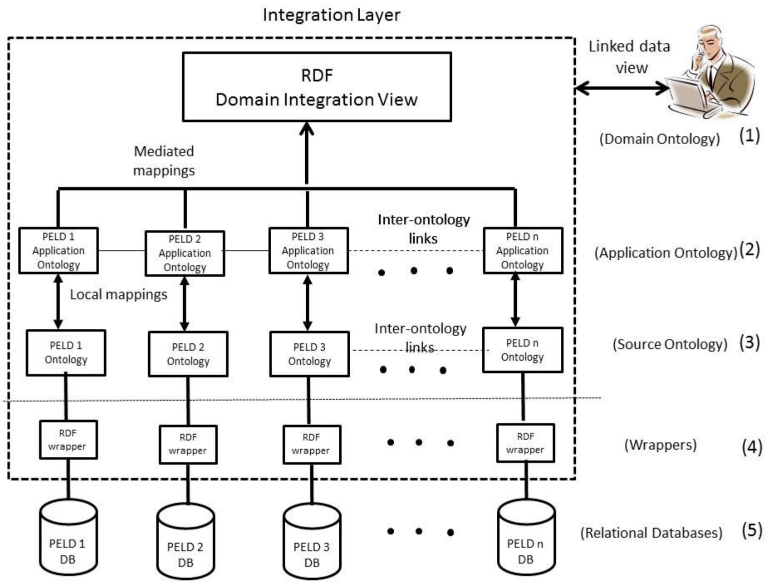

Figure 1. Three-level architecture for Linked Data Integration

The RDF Domain Integration View (1) is the Domain Ontology (DO) that represents the mediated schema. Designed by an expert user, it provides a conceptual representation of a specific domain, which comprises a global shared vocabulary and constraints. Each PELD relational database (5) is transformed into RDF by a specific wrapper (4) (see section 4.4) and becomes a source ontology (3), which is then rewritten as a PELD application ontology (APO) (2). It is worth observing that each APO describes a source ontology according to the principles of LD, which is a subset of the DO ontology. Application ontologies help breaking the query answering problem in two steps: (i) a query is submitted to the mediated schema, i.e., to the domain integration view, and by using mediated mappings, the query over the integration view is rewritten in terms of the application ontologies. As an example, consider queries over the Sample concept (Figure 2), which are rewritten as unions of AO; then (ii) based on the rewritten query an execution plan is generated, in which references between APOs become joins, and each sub-query, completely covered by an AO is rewritten using local mappings, and then submitted to the corresponding PELD local databases to retrieve information and deliver an integrated query answer to the user as LD. This step by step procedure is better described in section 6.

4. Application Scenarios

This section describes the Guanabara PELD scenarios that will be used for integration, based on the architecture depicted in Figure 1.

Guanabara PELD aims at getting biotic data from samples extracted from the bay water and from fishing resources. The living organisms are hierarchically classified in a taxonomy. The first level corresponds to the *Kingdom*, which is decomposed into *Phylums* and successively into *classes*, *orders, families*, *genders* and *species*. Each level has respectively its own subdivisions. Any level within this classification is called a *taxon*. There exist differences in the levels concerning each organism. Some of them have been reclassified, and in this case, both classifications are kept, and a synonymous relation is established between them.

In the context of ecological data analysis, some important features deserve some attention. Geographical region information identifies a target ecosystem and is used for selecting and classifying events according to their location. On the other hand, trophic relations are fundamental for the ecosystem study. Finally, the taxonomy enables a hierarchical analysis of the species. The analysis of these aspects may be explored by the use of inference in an integrated way.

The main characteristics of each scenario are described next.

- *Plankton:* in the plankton scenario, a sample data takes into account temporal (data and time) and spatial (latitude, longitude and profundity) information, as well as methods used for sample collect and conservation, atmospheric, and maritime conditions during each collect. For each analysis performed, data, sample and the applied method are registered. Biomass measurements of organisms found in the samples can be done at specie level or at the taxonomy highest level;
- *Community Fish:* besides temporal and spatial information, this application scenario stores the *fishing method* used to catch fishes, taking into account two different depths (initial and final). It is worth observing that collected fishes are divided into three samples, from which the total weight and number of individuals are analyzed for each *taxon* found in the collect process;
- *Catfish Genidens:* differently from the previous scenarios, this application scenario analyzes each specific specie individually, considering not only spatial and temporal references, but also the *fishing method* employed in the collect process, the specie *weight, length* and *gender*.

5. Domain and Application Ontologies

Based on the application scenarios described above, this section describes the ontologies generated at each level of the framework architecture presented in Figure 1.

Domain Ontology(DO)

Since in this paper the main purpose is not ontology design, we assume the domain ontology is provided by the user. Figure 2 presents the conceptual representation of the PELD domain ontology, referenced in our architecture as RDF Domain Integration

View. The namespace prefix "d" is used to refer to the vocabulary of this domain ontology. Since most of the class properties are self-described, we just give a few examples of the class properties. Thus, *d:collect_method* is defined as a *datatype property* with domain *d:Sample* and range *string; d:has_predator* is an object type property with domain *d:Trophic_Chain* and range *d:Taxon;* and *d:has_pl_analysis* is also defined as an object property, with domain *d:Plankton_sample* and range *d:Pl_analysis.*

Application Ontology(AO)

As mentioned in section 1, PELD sites are composed of different PELD subprojects. Each such PELD subproject takes part in the PELD data integration, by providing their local data published in RDF, which is rewritten as an AO, using a subset vocabulary of the DO. As in a federated database, an application ontology may be seen as an external ontology that takes part in the integrated schema, i.e., the domain ontology. Figure 3 presents a conceptual representation of the PELD AOs associated with the application scenarios described above comprising five ontologies: *Plankton, Catfish Genidens*, *Community Fishes, Region and Taxon*, each one having the following namespace prefixes: "apl:", "acf:","aco:", "r", and "tx" respectively. As mentioned before, the vocabulary of an application ontology consists of classes and properties that are subset of the domain ontology. Thus, access to the local data is done through direct mappings and the integration work becomes facilitated.

Based on the work proposed in [Vidal et al 2011], Figure 4 presents the list of the rules defined for the mapping between the APO and the DO. Due to space restriction we present only the mapping rules of *Plankton* ontology and we refer the reader to the above reference for more details on the definition of these rules, which is not in the scope of this paper.

It is worth mentioning that since the ontologies Region and Taxon represent data that are not frequently changed, they are previously materialized and stored locally as RDF triples in a repository, also as AOs. Thus, they are accessed whenever required and joined together with the other APOs that are virtually retrieved, as described in section 6.2.

6. Querying over the Framework Architecture

The main purpose of the proposed integration framework architecture is to answer user´s queries in terms of a domain ontology. Through the unified view exposed by the DO, researchers can access PELD subproject data transparently independently of local particularities. In order to deliver data, the data integration framework must be supported by a data integration engine that processes user´s query requests and returns results dealing with necessary data translations and access to source data[11]. In the context of this paper, ontologies in all architecture levels are homogeneously expressed in RDF. Thus, user requests may be submitted to the data integration system using SPARQL. The query expression is transformed into sub-queries over the application ontologies exposed as RDF triples by the D2RQ [Bizer et al. 2006] engine from the source databases.

The QEF system developed at DEXL laboratory has been used as the data integration engine. QEF is an extensible query engine that supports user-defined

[11] In the current version QEF does not rewrite queries yet. This is considered as a future work.

algebras and data structures. In order to support PELD data integration, a new version named QEF-LD [Magalhães 2012] has extended QEF. This new version includes linked data algebraic operators, and wrappers that submit AO sub-queries to a D2R endpoint. The latter exports local databases as virtual AOs.

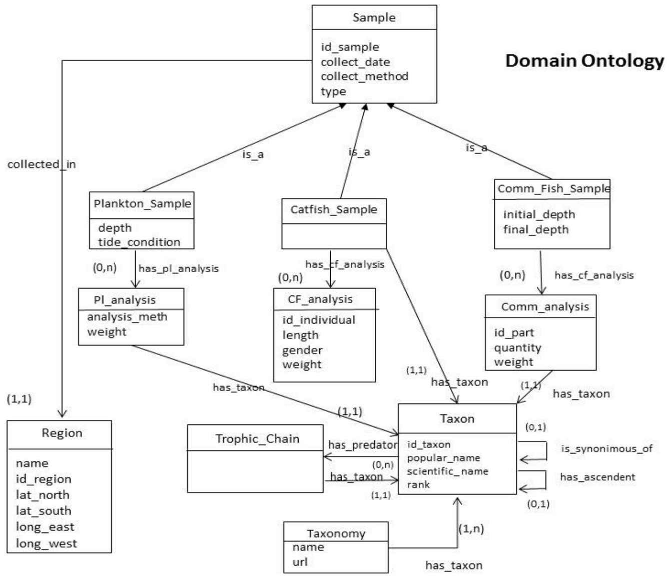

Figure 2. PELD domain ontology

In scenarios where a domain ontology query is translated into sub-queries over more than one application ontology, results are combined by the Union operator and returned to the user in a single result set.

Considering the strategy developed in [Vidal et al 2011], the following algorithm is performed:

- The user submits a SPARQL query to the data integration system expressed in terms of a domain ontology. Then, according to the mediated mappings, an integrated query execution plan is generated according to the following steps:
  a. References to the concepts Region and Taxonomy in the query, which are shared by the AOs, are mapped to BindJoins [Magalhães 2012] between the source AO and the shared databases (i.e. Region or Taxonomy).
  b. Each sub-query is submitted to a data source. D2R endpoints translate the submitted queries to the corresponding local database queries. The Region and Taxonomy AOs are materialized as RDF sources and joined with AO ontology concepts through SPARQL queries.
  c. Once the results are obtained, QEF applies the joins and unions handling in the final result. A query over the Sample concept is rewritten as Unions of subqueries over each APOs (see figure 4 (a)), according to the mappings presented in Figure 4(b), respectively. This step is not currently supported by QEF-LD [Magalhães 2012].

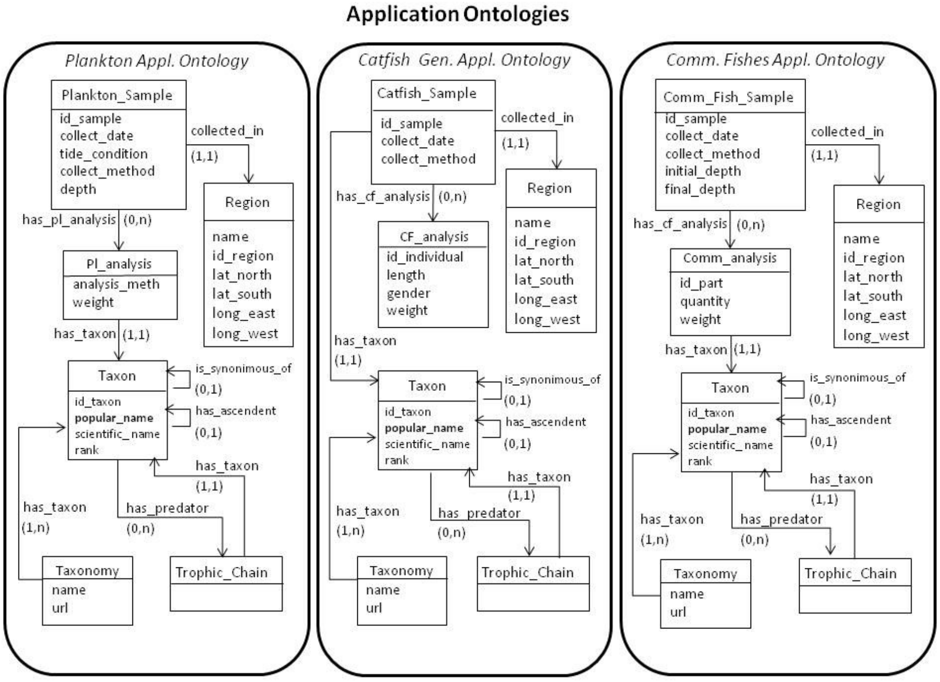

Figure 3. PELD application ontologies

d:Sample(p) ⇐ apl: Plankton_Sample(pls) U acf: Catfish_Sample(c) U
aco: Comm_Fish_Sample(co)

Figure 4(a). Sample DO expressed as the union of the different sample species of APO ontologies

1. d:Pl_analysis(pl) ⇐ apl:Pl_analysis(pl)
2. d: Plankton_Sample(pls) ⇐ apl: Plankton_Sample(pls)
3. d:id_sample(p,id) ⇐ apl: id_sample(pls,id), apl: Plankton_Sample(pls)
4. d:col_date(p,dt) ⇐ apl: collect_date(pls,dt), apl:Plankton_Sample(pls)
5. d:collect_method (p,cm) ⇐ apl: collect_method(pls,cm), apl: Plankton_Sample(pls)
6. d:depth (pls,d) ⇐ apl:depth(pls,d), apl: Plankton_Sample(pls)
7. d:tide_condition (pls,tc) ⇐ apl: tide_condition(pls,tc), apl: Plankton_Sample(pls)
8. d:collect_in (p,l) ⇐ apl: collect_in(pls,l), apl:Plankton_Sample(pls), Region(l)
9. d:analysis_meth (pl,am) ⇐ apl: analysis_meth(pl,am), apl:Pl_analysis(pl)
10. d:weight (pl,w) ⇐ apl: weight(pl,w), apl:Pl_analysis(pl)
11. d:has_taxon (pl,tx) ⇐ apl:has_taxon(pl,tx) , apl:Plankton_analysis(pl), Taxon(tx)
12. d:has_pl_analysis (p,pl) ⇐ apl:has_pl_analysis(pls,pl)
13. d:type (p,´plankton`) ⇐ apl: Plankton_Sample(p)

Figure 4(b). Mapping rules from the Plankton APO to DO

6.1 Submitting a Query

According to the strategy presented above, the following query has been submitted to the proposed framework: "*Get the species found at Paquetá Island in 2004, their synonyms and predators*". In the following paragraphs the transformation process for answering this query is described, step by step.

i) The main query (Q) is expressed in terms of the domain ontology, which comprises the union of the 3 species: Planktons, Catfish and Comm. Fish.

Select distinct ?name ?name_syn
?name_pred
Where {
{
?s d:collected_in ?r .
?s d:collect_date ?dt.
?r d:name ?reg.
?p d:is_a ?s .
?p d:has_pl_analysis ?pl .
?pl d:id_taxon ?tx.
?tx d:popular_name ?name
}
Union {
?p d:is_a ?s .
?cf d:id_taxon ?tx .
?tx d:popular_name ?name .
}
	Union {
	?p d:is_a ?s .
	?p d:has_cf_analysis ?pl .
	?pl d:id_taxon ?tx .
	?tx d:popular_name ?name .
	}
	Optional {
	?tx d:has_predator ?pred .
	?pred d:has_taxon ?idpred .
	?idpred d:popular_name ?name_pred .
	}
	Optional {
	?syn d:is_synonimous-of ?tx;
	d:popular_name ?name_syn .
	}
	Filter (?reg = "Paqueta" && ?dt = 2004)
	}
	order by ?name

ii) Query Q is rewritten as the union of three subqueries Q_1, Q_2 and Q_3, which aim at extracting data from *Plankton, Catfish Genidens,* and *Comm. Fish* application ontologies, region and taxon, respectively (Figure 5).

6.2 Executing a Query in QEF

As mentioned before, part of the application ontologies are stored in RDF tuples as materialized views. Such characteristic requires an execution plan for each query Q_i (Figure 6(a)), in order to guide QEF into the correct execution of the algebra operators sequence over the local data sources.

In order to exemplify this step, consider query Q'_1 the Q_1 version about *Planktons* that will be submitted to QEF. Similarly to Q_2 and Q_3, these queries use both virtual and materialized information. In other to describe the step by step execution procedure performed by QEF, Figures 6 (a) and (b) present respectively a Q'_i query execution plan for each Q_i, and each corresponding SPARQL query. Figures 7, 8, and 9 present, respectively, the results of Q´1, Q´2 and Q´3.

Final results (Figure 10) are obtained from Q´1 ∪ Q´2 ∪ Q´3, having duplicated values discarded.

7. Conclusion

This paper reports on the application of the aforementioned data integration framework to the ecological domain and the extension of QEF, a data integration system, to answer queries on heterogeneous ecological databases using this framework. A complete data integration scenario is discussed based on the challenges involved in publishing ecological data produced by the PELD Guanabara project, in Brazil.

Based on the data integration framework, a set of PELD subproject databases stored in relational databases are transformed into RDF as endpoints via D2RQ, which enable an integrated view over the data resources via SPARQL queries. The results indicate that the proposed data integration framework is promising and that shall be adopted as a standard for more complex ecological database integration scenarios.

Q_1	Q_2	Q_3

Select distinct ?name
?name_syn ?name_pred
Where {
?s apl:collected_in ?r.
?s apl:col_date ?dt.
?r r:name?reg.
?s apl:has_pl_analysis ?a.
?a tx:has_taxon ?tx.
?tx tx:popular_name
?name.
Optional {
?tx tx:has_predator ?pred.
?pred tx:has_taxon ?idpred.
?idpred tx:popular_name
?name_pred.
}
Optional {
?syn tx:is_synonimous_of
?tx;
tx:popular_name
?name_syn}.
Filter (?reg = "Paqueta" &&
?dt = 2004).
} order by ?name
	Select distinct ?name
	?name_syn ?name_pred
	Where {
	?s acf:collected_in?r.
	?s acf:col_date ?dt.
	?r r:name ?reg.
	?s tx:has_taxon ?tx.
	?tx tx:popular_name ?name.
	Optional {
	?tx tx:has_predator ?pred.
	?pred tx:has_taxon ?idpred .
	?idpred tx:popular_name
	?name_pred.
	}
	Optional {
	?syn tx:is_synonimous_of
	?tx;
	tx:popular_name ?name_syn
	}
	Filter (?reg = "Paqueta" &&
	?dt = 2004).
	}
	order by ?name
		Select distinct ?name ?name_syn
		?name_pred
		Where {
		?s aco:collected_in ?r.
		?s aco:col_date ?dt.
		?r aco:name?reg.
		?s aco:has_cf_analysis ?a.
		?tx tx:popular_name ?name.
		?a tx:has_taxon ?tx.
		Optional {
		?tx tx:has_predator ?pred.
		?pred tx:has_taxon ?idpred.
		?idpred tx:popular_name
		?name_pred.
		}
		Optional {
		?syn tx:is_synonimous_of ?tx;
		tx:popular_name ?name_syn
		}
		Filter (?reg = "Paqueta" && ?dt
		= 2004).
		}
		order by ?name

Figure 5. Q_i SPARQL query

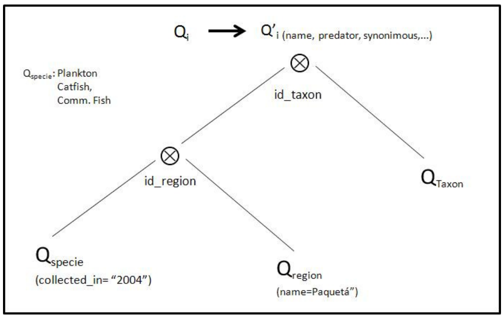

Figure 6(a). Q'_i execution plan

Q'_1 (Planktons)	Q'_2 (Catfish)	Q'_3 (Comm.fish)

$Q_{plankton}$: Select ?id_taxon,
?id_region
Where {
?s apl:collected_date ?dt.
?s apl:collected_in ?id_region.
?s apl:has_pl_analysis ?id_an.
?id_an tx:has_taxon ?id_taxon.
Filter (?dt =2004).
}
-- 	
Q_{region}: Select ?id_region
Where {
	$Q_{catfish}$: Select ?id_taxon,
	?id_region
	Where {
	?s acf:collected_date ?dt.
	?s acf:collected_in ?id_region.
	?s tx:has_taxon ?id_taxon.
	Filter (?dt =2004).
	}
	-- 	
	Q_{region}: Select ?id_region
	Where {
		$Q_{commfish}$: Select ?id_taxon,
		?id_region
		Where {
		?s aco:collected_date ?dt.
		?s aco:collected_in ?id_region.
		?s aco:has_cf_analysis ?id_an.
		?id_an tx:has_taxon ?id_taxon.
		Filter (?dt =2004).
		}
		-- 	
		Q_{region}: Select ?id_region
		Where {

?id_region r:name ?n.
?r r:id_region ?id_region.
Filter (?n, “Paquetá”).}
-- 	
Q_{Taxon}: Select distinct ?name ?name_syn ?name_pred
Where {
?x tx:id_taxon ?id_taxon.
?id_taxon tx:scientific_name ?name.
?id_taxon tx:has_predator ?pred.
?pred tx:has_taxon ?tx_pred.
?tx_pred tx:scientific_name ?name_pred.
?id_taxon tx:is_synonimous_of ?syn_tax.
?syn_tx tx:scientific_name ?name_syn.
}
	?id_region r:name ?n
	?r r:id_region ?id_region.
	Filter (?n, “Paquetá”).}
	-- 	
	Q_{Taxon}: Select distinct ?name ?name_syn ?name_pred
	Where {
	?x tx:id_taxon ?id_taxon.
	?id_taxon tx:scientific_name ?name.
	?id_taxon tx:has_predator ?pred.
	?pred tx:has_taxon ?tx_pred.
	?tx_pred tx:scientific_name ?name_pred.
	?id_taxon tx:is_synonimous_of ?syn_tax.
	?syn_tx tx:scientific_name ?name_syn.
	}
		?id_region r:name ?n
		?r r:id_region ?id_region.
		Filter (?n, “Paquetá”).}
		-- 	
		Q_{Taxon}: Select distinct ?name ?name_syn ?name_pred
		Where {
		?x tx:id_taxon ?id_taxon.
		?id_taxon tx:scientific_name ?name.
		?id_taxon tx:has_predator ?pred.
		?pred tx:has_taxon ?tx_pred.
		?tx_pred tx:scientific_name ?name_pred.
		?id_taxon tx:is_synonimous_of ?syn_tax.
		?syn_tx tx:scientific_name ?name_syn.
		}

Figure 6(b). Q´i SPARQL query

SPARQL results:

name	name_syn	name_pred
"Acartia tonsa"	"Acartia (Acanthacartia) tonsa"	"Argentine menhaden"
"Acartia tonsa"	"Acartia (Acanthacartia) tonsa"	-
"Acartia tonsa"	"Acartia (Acanthacartia) tonsa"	"Marinis anchovy"
"Oithona plumifera"	-	-
"Oithona plumifera"	-	"Argentine menhaden"
"Oithona plumifera"	-	"Marinis anchovy"
"Oitona hebes"	-	-
"Oitona hebes"	-	"Argentine menhaden"
"Oitona hebes"	-	"Marinis anchovy"
"Temora turbinata"	-	-
"Temora turbinata"	-	"Argentine menhaden"
"Temora turbinata"	-	"Marinis anchovy"

Figure 7. Results of Q´1

SPARQL results:

name	name_syn	name_pred
"Marine catfish"	-	-

Figure 8. Results of Q´2

SPARQL results:

name	name_syn	name_pred
"Argentine menhaden"	-	-
"Argentine menhaden"	-	"Marine catfish"
"Argentine menhaden"	-	"Whitemouth croaker"
"Franciscana"	-	-
"Marine catfish"	-	-
"Marinis anchovy"	-	-
"Marinis anchovy"	-	"Marine catfish"
"Marinis anchovy"	-	"Whitemouth croaker"
"Marinis anchovy"	-	"Franciscana (Stenodelphis)"
"Whitemouth croaker"	-	-
"Whitemouth croaker"	-	"Franciscana (Stenodelphis)"

Figure 9. Results of Q´3

Num	Name	Synonym	Predator Name
1	Marine catfish	null	null
2	Acartia tonsa	Acartia (Acanthacartia) tonsa	Argentine menhaden
3	Acartia tonsa	Acartia (Acanthacartia) tonsa	null
4	Acartia tonsa	Acartia (Acanthacartia) tonsa	Marinis anchovy
5	Oithona plumifera	null	nul
6	Oithona plumifera	null	Argentine menhaden
7	Oithona plumifera	null	Marinis anchovy
8	Oitona hebes	null	null
9	Oitona hebes	null	Argentine menhaden
10	Oitona hebes	null	Marinis anchovy
11	Temora turbinata	null	null
12	Temora turbinata	null	Argentine menhaden
13	Temora turbinata	null	Marinis anchovy
14	Argentine menhaden	null	null
15	Argentine menhaden	null	Marine catfish
16	Argentine menhaden	null	Whitemouth croaker
17	Franciscana	null	null
18	Marine catfish	null	null
19	Marinis anchovy	null	null
20	Marinis anchovy	null	Marine catfish
21	Marinis anchovy	null	Whitemouth croaker
22	Marinis anchovy	null	Franciscana (Stenodelphis)
23	Whitemouth croaker	null	null
24	Whitemouth croaker	null	Franciscana (Stenodelphis)

Figure 10. Final Results

ACKNOWLEDGEMENTS

This work has been partially supported by CNPq through its Institutional Capacity Program (Proc. 382.489/09-8) and Productivity Research fellowship (Proc. 309502/2009-8).

References

Bizer C., Health T., Berners-Lee T. D2R Server – Publishing relational databases on the Web as SPARQL endpoints. Proc. of the 15th International World Wide Web Conference, Edinburgh, Scotland, 2006.

Campos, S.R., Martinhago A.Z., Massahud R.T., França A.M., Prieto L. E., Mendes J.D.C. Database modeling of the economic ecological zoning of Minas Gerais using UML-GeoFrame (in Portuguese). Proc. of the XIV Brazilian Symposium of Remote Sensoring, Natal, Brasil, 25-30 April, 2009, INPE, p. 4943-4949.

Cavalcanti, M. J. Database on Amazon biodiversity: experience on the Biotupé project. Biotupé: Physical environment, biological diversity and sociocultural of Low Negro River, central Amazon (*in Portuguese*). Santos-Silva, Aprile, Scudeller, Editora INPA, Manaus, 2005.

Gray A. J. G., Gray N., Ounis I. Can RDB2RDF tools feasibily expose large science archives for data integration? The Semantic Web: Research and Applications – LNCS, Vol. 5554/2009, 491-505, 2009. DOI: 10.1007/978-3-642-02121-3_37.

Heath, T., Bizer C. Linked Data: evolving the Web into a global data space (1st edition). Synthesis lectures on the semantic Web: theory and technology, 1:1, 1-136. Morgan & Claypool ed., 2011.

Langegger, A., Wöß, W., Blöchl, M. 2008. A Semantic Web Middleware for Virtual Data Integration on the Web. In: Proceedings of the 5th European Semantic Web Conference (ESWC). Volume 5021 of Lecture Notes in Computer Science. Springer Verlag, pp. 493–507.

Magalhães, R. P. Um Ambiente para Processamento de Consultas Federadas em Linked Data Mashups. M.S. thesis, Universidade Federal do Ceará, 2012.

Manola, F. and Miller, E. RDF primer. W3C Recommendation, February, 2004. Available at: http://www.w3.org/TR/rdf-primer.

Manzi, A. Data management of Brazilian long-term ecological research projects (*in Portuguese*), research project, Edital MCT/CNPq Nº 59/2009 – PELD support proposals, 2009.

Porto F., Tajmouati O., Silva V. F. V., Schulze B., Ayres F. V. M. QEF - supporting complex query applications, 7th IEEE International Symposium on Cluster Computing and the Grid — CCGrid 2007 , Rio de Janeiro, Brazil, pp. 846-851.

Prud´hommeaux, E. and Seaborne, A. 2008. Sparql Query Language for RDF. W3C Recommendation. Available at: http://www.w3.org/TR/rdf-sparql-query/.

Prud'hommeaux, E. And Buil-Aranda, C. SPARQL 1.1 Federated Query. http://www.w3.org/TR/sparql11-federated-query/, 2011.

Quilitz, B. and Leser, U. 2008. Querying Distributed RDF Data Sources with SPARQL. In: Proceedings of the 5th European Semantic Web Conference (ESWC). Volume 5021 of Lecture Notes in Computer Science, Springer Verlag, pp. 524–538 (2008).

Schultz, A., Matteini, A., Isele, R., Bizer, C., and Becker, C. LDIF : Linked Data Integration Framework. In Proceedings of the 11th Interational Semantic Web Conference ISWC2011. pp. 1–4, 2011.

Schwarte, A., Haase, P., Hose, K., Schenkel, R., and Schmidt, M. Fedx: optimization techniques for federated query processing on linked data. In Proceedings of the 10th international conference on The semantic web - Volume Part I. ISWC'11. Springer-Verlag, Berlin, Heidelberg, pp. 601–616, 2011.

Vidal, V.M.P, Macêdo, J.A.F., Pinheiro, J. C., Casanova, M. A., Porto F. Query processing in a mediator based framework for linked data integration. IJBDCN 7(2): 29-47, 2011.