Ecologically-aware Queries for Biodiversity Research

Luiz Celso Gomes Jr[1], **Claudia Bauzer Medeiros**[1]

[1]Instituto de Computacao – UNICAMP
Caixa Postal 6176 – 13081-970 – Campinas – SP – Brazil

{luizcelso,cmbm}@ic.unicamp.br

Abstract. *To carry ecologically-relevant biodiversity research, one must collect chunks of information on species and their habitats from a large number of institutions and correlate them using geographic, biologic and ecological knowledge. Distribution and heterogeneity inherent to biodiversity data pose several challenges, such as how to find and merge relevant information on the Web, and process a variety of ecological and spatial predicates. This paper presents a framework that exploits advances in data interoperability and Semantic Web technologies to meet these challenges. The solution relies on ontologies and annotated repositories to support data sharing, discovery and collaborative biodiversity research. A prototype using real data has implemented part of the framework.*

1. Introduction

Biodiversity is an outstanding example of a scientific domain that deals with heterogeneous datasets and concepts from many areas. Biodiversity studies rely on models to define species richness, abundance, endemism, distribution and so forth. To create the models, species occurrence data must be obtained from diverse institutions, and be combined with other kinds of data, such as phylogenetic data (describing evolutionary relations), taxonomic data for nomenclature, data describing ecological correlations among species and geographic data depicting habitat conditions.

Typically, biodiversity information systems provide support to queries that are centered on the so-called *collection* or *occurrence* records, managed by museums or by research institutions. An occurrence record stores data on some kind of observation of living beings – it includes data on species' taxonomic classifications, location where the species were observed or collected, by whom, when and how. Additional data sources include geographical data (e.g. on habitats, or climate variables), and several kinds of annotations. The most common queries on such systems concern species' spatial distribution in a given area. Other queries may demand sets of occurrence records that satisfy a given predicate, or computation of aggregate functions over such records. Scientists may also want to find out more about specific geographic areas (e.g., rainfall or temperature patterns), thereby being able to compute climate models, or run simulations on habitat variables.

Query predicates, in these systems, can be classified into two categories: those that involve operations that are typically computed by standard DBMS mechanisms and those that involve computing spatial predicates. The latter either requires extended DBMS capability e.g., using PostGIS or, more commonly, a GIS. Thus, end-user requests in a typical biodiversity information system are solved by combining spatial correlations to functions used in a DBMS. This, however, only supports a subset of the functionality demanded by bio-scientists.

These end-users also need more complex computations, e.g., requiring spatio-temporal query processing, such as deriving co-occurrence of species in a given space-time frame. Such processing is seldom supported. Other predicates involve ecological relations among species, e.g., predator-prey or parasitic relationships. Such relationships are not stored, and must be deduced by the scientist after performing a sequence of queries and simulations. Most times, scientists have to invest a considerable amount of time, and perform many manual tasks, to obtain the needed data.

This paper proposes a framework to fill this gap. Besides supporting the more usual kinds of query predicates, it also allows computation of ecological predicates, by combining stored and derived data and ontologic information, for distributed data repositories. This framework has been partially implemented using data from the Institute of Biology, UNICAMP, within an eScience biodiversity project[Medeiros et al. 2007].

The main contributions of the paper are, therefore: (i) the specification of a framework that allows scientists to pose semantically rich queries, encompassing taxonomic, ecological and geographic predicates and (ii) the validation of the framework by the partial implementation of a prototype, using real data.

2. Related work

2.1. Biodiversity research

Research in biodiversity is devoted to understanding the diversity of life and trying to find ways to preserve it. Biodiversity is, however, a complex subject. To begin with, estimates for the total number of species in the planet range to up to 80 million [Wilson 1999] — the bulk of this amount yet to be discovered. Moreover, to undertake biodiversity studies, scientists have to take into account species interactions, both among species and with their environment.

The major interactions between *pairs* of species include competition, predation and mutualism [Morin 1999]. Many more complex interactions can be derived from these elementary processes. Food chains, for example, are pathways of nutrient flow through a sequence of species arranged according to their predator-prey interactions. Another important concept in ecological research is that of *taxonomic relations*, which forms the foundation that enables scientists to properly interpret each other's work [Wilson 1999].

Species interactions with the environment are assessed by combining geographic and ecological data. Therefore, finding and accessing geographical data becomes critical in biodiversity research [Guralnick and Neufeld 2005]. Geographic constraints related to natural conditions (e.g. climate and relief) and human activities (e.g pollution) have direct impact in species richness and distribution. *Species occurrence* data, which also contains geographic information, is the basic unit of information for biodiversity measurements, as mentioned in Section 1. They allow studies on species distribution patterns, thereby supporting efforts on conservation initiatives.

2.2. Biodiversity data sharing

Work on biodiversity involves scientists from many fields, and requires combining a variety of distributed heterogeneous data sources on the Web. Geospatial Web services and exchange standards for occurrence records are important elements in promoting biodiversity data integration and interoperability among systems.

Data sharing and integration is often based on geographic coordinates. Thus, geospatial Web services are considered in many solutions [Guralnick and Neufeld 2005]. The Open Geospatial Consortium (OGC) [Open Geospatial Consortium Inc. (OGC)] is an international organization that leads the development of standards for interoperability among geospatial applications. The consortium defines the Web Feature Service (WFS) [OGC 2005b] specification to provide a standardized means to access geospatial data encoded in the Geographic Markup Language (GML) [OGC 2003]. GML, also from OGC, is an XML-based standard for the transport and storage of geospatial information. The WMS (Web Map Service) specification defines means to produce two-dimensional maps from geospatial data.

There are many initiatives to leverage sharing and interoperability of species occurrence data. Darwin Core [Taxonomic Databases Working Group (TDWG)] is an XML-based standard that defines the necessary elements to describe species occurrence data, constituting the first step towards data interoperability. Infrastructures for sharing biodiversity data on the Internet (such as Species Analyst [Species Analyst project]) rely on exchange standards and transmission protocols to build an interconnected network of data providers. A scientist interacts with such systems by indicating target sites and data sources and posing queries through a standard interface. Queries are usually limited to textual predicates and return raw occurrence records to the user. Such infrastructures do not allow more elaborate queries, and it is up to the scientists to perform any kind of semantic post-processing.

2.3. Ontologies

Ontologies are being used in Computer Science to formalize shared conceptualizations within communities. An ontology organizes concepts to convey semantic information and to allow new knowledge to be inferred [Gruber 1995].

The Semantic Web initiative is pushing forward the use of ontologies to provide the Web with a machine-understandable metadata framework, fostering interoperability. The World Wide Web Consortium (W3C) is the main player in the Semantic Web initiative. W3C specified the Web Ontology Language (OWL) [Daconta et al. 2003], the standard for ontology specification. OWL is based on the Resource Description Framework (RDF) [Daconta et al. 2003], which is a general-purpose language to represent and correlate Web resources.

W3C is developing the SPARQL query language for querying RDF data [Seaborne and Prud'hommeaux 2006]. A SPARQL query is formatted in terms of RDF triple patterns. Queries are evaluated via pattern matching between the query expression and the RDF graph.

Many biodiversity projects have begun to explore the use of ontologies to allow data sharing on the Web. The SPIRE project [Parr et al. 2006] is investigating how Semantic Web technologies can be applied to the biodiversity domain. The project is developing ontologies for taxonomic, ecological and niche modeling concepts, and is producing tools based on the ontologies. Among the tools is an on-line query form that allows users to submit SPARQL queries. Query results return fragments of the ontologies, expressed in OWL. There is no attempt to retrieve other kinds of biodiversity-related data available in Web repositories.

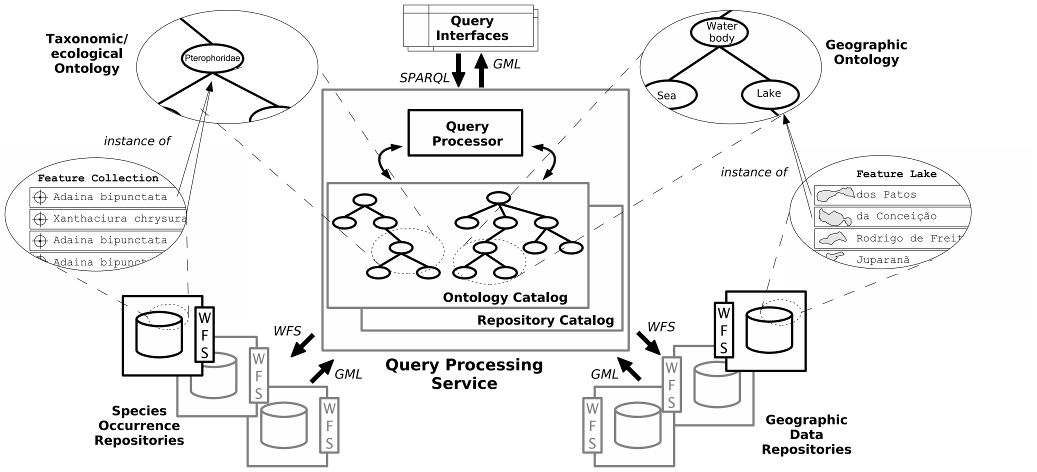


Figure 1. Overview of the interactions among the architecture's elements

3. Ecologically-aware queries

This section presents the architecture of our infrastructure for processing ecological queries. It integrates all trends presented in the previous section: it employs (i) domain ontologies to provide a global model of the data to be shared, (ii) standards to access remote data repositories, and (iii) a combination of spatial, textual and ecological predicates to process ecologically-aware queries.

3.1. Architecture overview

The architecture is composed of three elements: (i) query interfaces, where users pose biodiversity queries, (ii) a query processing service, that processes queries received from the interfaces and (iii) distributed repositories, from where the query service retrieves data. Figure 1 presents a high level view of these elements and their interactions. Query interfaces are applications tailored to specific goals (e.g., predict species occurrence, establish conservation priorities) and users (e.g., biologist, ecologist). User queries at the interface are translated to SPARQL and forwarded to the query processing service.

This query processing service (center of the figure) is the main element of the architecture. Its role is to disambiguate predicates with help of ontologies, to find the appropriate data in distributed Web repositories, process these data, and return the results to the users. The repositories (left and right bottom) are databases published by research groups and institutions. There are two types of repositories: those that hold occurrence records, and those that hold data on geographic objects such as lakes, countries or biomes. The figure shows examples of data published by the institutions. Occurrence and geographic data records are georeferenced (i.e. associated with geographic coordinates).

The figure also shows that the query processor makes extensive use of ontologies to expand terms and to process predicates. The ontology on the left contains taxonomic and ecological information. Its expanded view shows the *Tephritidae* concept (the family of insects that includes fruit flies). The ontology on the right contains geographic information, with *Water Body* and *Lake* concepts in the expanded view. As shown by the arrows among these detailed views, repositories' contents are associated - in a conceptual level - with ontology elements.

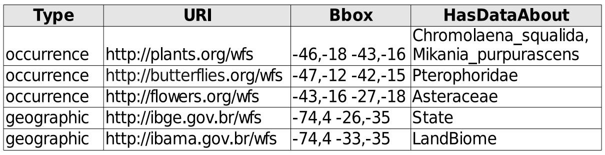
3.2. The query processing service

The query service is composed of a query processor and catalogs. The **query processor** – see Figure 1 – receives SPARQL queries from query interfaces (whose design is outside the scope of this paper). The processor's output is a GML file that can be used to generate maps at the interfaces.

Query processing requires internal data structures, stored in **catalogs**. The term "catalog" was adopted to establish an analogy with standard DBMS query processing mechanisms, where catalogs store information such as database schemas or data allocation properties [Elmasri and Navathe 1994]. The service's catalogs are used by the processor in tasks such as expanding query terms and finding target repositories. There are two kinds of catalog: Domain Ontology Catalog and Repository Catalog. Their contents are expected to be consistent – i.e., there is no conflicting information.

The **Domain Ontology Catalog** stores the ontologies containing taxonomic/ecological and geographic concepts. Its content is provided by research communities. It is used by the query processor to expand queries and process ecological predicates. The taxonomic/ecologic ontology contains assertions such as "*Adaina bipunctata* (a butterfly species) is a subclass of *Pterophoridae* (a family) that preys on plant species *Chromolaena squalida*". The geographic ontology holds taxonomic classifications of geographic phenomena, such as "concept *Lake* is-a *Water Body*".

The **Repository Catalog** plays the role of an "index" to biodiversity data sources on the Web. It contains entries registered by trusted institutions and research groups. As depicted in Figure 2, each such entry is composed of four main fields: the repository type, its URI, a geographic bounding box, and a set of ontologic annotations from the Ontology Catalog. The *type* field indicates whether the Web repository contains information on occurrence or geographic phenomena. The *bounding box* defines the geographic region for which the repository can provide data. The ontologic annotations qualify the contents of a repository. Repository registering assumes that occurrence data records are compliant to the Darwin Core standard [Taxonomic Databases Working Group (TDWG)]. All repositories must be compliant with the WFS standard, thus standardizing interfaces and providing means to apply geographic filters in data retrieval.



Type	URI	Bbox	HasDataAbout
occurrence	http://plants.org/wfs	-46,-18 -43,-16	Chromolaena_squalida, Mikania_purpurascens
occurrence	http://butterflies.org/wfs	-47,-12 -42,-15	Pterophoridae
occurrence	http://flowers.org/wfs	-43,-16 -27,-18	Asteraceae
geographic	http://ibge.gov.br/wfs	-74,4 -26,-35	State
geographic	http://ibama.gov.br/wfs	-74,4 -33,-35	LandBiome

Figure 2. Entries in the repository catalog

3.3. Query processing

Figure 3 shows the sequence of phases in query processing. The processor receives an extended SPARQL query (Phase A) and returns a GML file containing the desired data (Phase C).

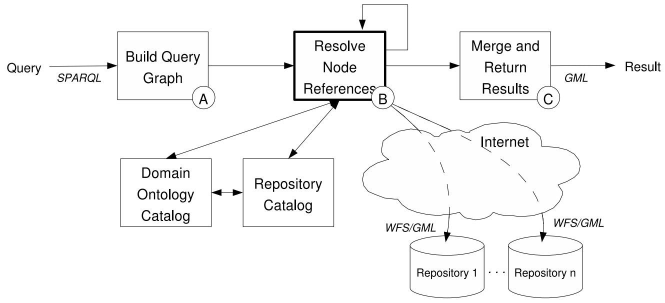


Figure 3. Query processing phases

The framework is strongly based on ontology processing. Ontologies and their elements intervene at each step of query processing. For this reason, the solution for query execution favors structures to process ontologies – i.e., all intermediate results are used to create, match and expand graphs. The three main phases are:

> **A) Build Query Graph:** Analyse the input query, and build the corresponding graph. The graph generated is a straightforward materialization of the graph implicitly expressed in the query: in a query graph $G(V, E)$ for a query Q, (i) $u \in V \Leftrightarrow u$ is subject or predicate of Q and (ii) $(u, v) \in E \Leftrightarrow$ there is a predicate in Q associating the subject of u and the object of v. The graph's vertices and edges are labeled with the URIs expressed in the SPARQL query.
>
> **B) Resolve Node References:** Iteratively process the query graph, resolving undefined elements. First, the framework's internal catalogs are checked; next, WFS requests are sent to the appropriate Web repositories to retrieve records. The result is a graph, or set thereof, extended with data retrieved from the repositories.
>
> **C) Merge and Return Results:** Process the contents of the graph(s) resulting from phase B and translate them into GML. The resulting file is returned to the interface level.

Algorithm 1 Process leaf-branches

```
Require: query graph G
Ensure: All graph nodes are resolved
while G has leaf-branches to be resolved do
    b ⇐ highest priority unresolved branch
    if priority(b) = 1 then {b can be resolved locally}
        update query graph
    else if priority(b) = 2 then {b's resolution requires data from catalog}
        resolve using Ontology Catalog data
        apply results to the query graph, updating priorities
    else {b's resolution requires data from repositories}
        simplify spatial predicates
        determine repositories to query, using Repository Catalog
        assemble and submit WFS queries to repositories
        apply results to the query graph, updating priorities
    end if
end while
```

Step B is the most complex, and is subdivided into several steps according to Algorithm 1 (error conditions are omitted). We name a *leaf-branch* a set composed of one single-degree vertex (a leaf), its incident edge, and the edge's other vertex (hereafter referred to as the branch's *base*). More formally, a leaf-branch B in a query graph G(V,E) may be defined as

$$B = \{(u, v, (u, v)) : u, v \in V \wedge (u, v) \in E \wedge degree(u) = 1\}$$

The algorithm is applied iteratively to each leaf-branch of the graph until the graph is completely resolved. It is suitable to connected, acyclic graphs (trees). The resolution of a leaf-branch comprises analyzing the predicate expressed in the edge, processing this predicate according to the object encoded in the leaf, and applying the results to the branch base. For this reason, we employ the term branch (rather than just leaf) resolution, to indicate the processing to be performed. At the end of a branch processing, its leaf is eliminated and its base contains the results of the processing. The algorithm uses a table [Gomes Jr 2007] to assign priorities to each leaf-branch according to their type. Priority 1 (the highest priority) branches are resolved locally (only by rearranging the query graph), priority 2 branches need the ontologies in the catalog, and lower priority branches (3 and 4) need remote queries to repositories. The goal of this strategy is to postpone costly operations until there is more information to filter intermediate results, avoiding retrieval of unnecessary data. The key steps are described in the following. For more details, the reader is referred to [Gomes Jr 2007].

Obtain highest priority branch (line 2): Chooses one leaf-branch among those with the highest priority, which is to be resolved in the subsequent steps of the algorithm.

Update query graph (line 4): For priority 1 branches, the resolution consists of a simple manipulation in the query graph (e.g. pruning). These branches are handled first, since they do not demand processing data.

Resolve using Ontology Catalog data (line 6): For priority 2 branches, the resolution consists on getting the needed information from the Ontology Catalog.

Simplify spatial predicates (line 9): The resolution of branches with priority higher than 2 involves retrieving data from Web repositories. Whenever a branch bearing spatial predicates enters this step, these predicates can be pre-processed to simplify data retrieval e.g., redundant predicates can be excluded [Rodríguez et al. 2003]. A deeper study on which optimizations may be done in this step is still in progress. The subsequent steps of the algorithm consider that geographic predicates have been pre-processed to restrict the spatial extent of queries submitted to repositories.

Determine repositories to query (line 10): Checks the Repository Catalog for a list of repositories that may provide instances regarding the current branch. This is processed by matching the branch's contents with the type, ontologic annotations and eventually the bounding boxes in the Repository Catalog.

Assemble and submit WFS queries (line 11): Assembles WFS queries tailored to each repository identified in the previous step. Asynchronously submits these queries to the appropriate repositories.

Apply results to graph (lines 7 and 12): Translates into graph representation results from the queries to the Ontology Catalog or repositories. Updates priorities.

4. Example

Let us now consider the following query: "return all *occurrence records* of species that are *preyed on by* the species *Adaina Bipunctata* and have been found *in São Paulo State's Atlantic Rainforest Biome*". This query contains ecological (prey on), spatial (in) and taxonomic predicates (*species = Adaina Bipunctata*). Additional spatial predicates are defined by naming geographic areas (São Paulo, Atlantic Rainforest). The processing of taxonomic and ecological predicates is based on the ontologies. The processor deals with spatial relations by building geographic filters to retrieve data in the repositories.

```
PREFIX te: <http://. . ./webios/taxo_eco.owl#>
PREFIX geo: <http://. . ./webios/geographic.owl#>
PREFIX sr: <http://. . ./webios/spatial_relation.owl#>
SELECT ?occurrence
WHERE { te:Adaina_Bipunctata te:predatorOf ?species .
        ?occurrence a ?species .
        ?occurrence sr:within geo:Sao_Paulo .
        ?occurrence sr:within geo:Atlantic_Rainforest .
```

Figure 4. Example of query using SPARQL syntax

Figure 4 shows the corresponding query in syntax that is compatible with SPARQL. In the code, the prefixes *te* and *geo* respectively stand for the taxonomic/ecological and geographic domain ontologies, which are to be used to process the query. The prefix *sr* indicates spatial predicates. Accepted spatial expressions are those specified by OpenGIS Spatial Filter Implementation [OGC 2005a], themselves representing the standard binary relationships found in the literature (e.g., [Rodríguez et al. 2003] – such as within, overlaps or disjoint). Keyword *a* is the standard syntax for "instance of" relationships in SPARQL. SPARQL queries provide access to multiple name spaces via the FROM clause; however, all found examples in the literature (and in Web sites) pressupose that there is a possibility of constructing a single ontology graph to be queried from the name spaces. Also, they do not allow accessing multiple ontologies at a time. Thus, this request needs to be decomposed into several queries. To do this, we start by building a query graph.

Figure 5 shows intermediate states of the query graph during the processing of the example query. Figure 5(1) depicts the graph in the beginning of the first iteration of Algorithm 1, which is the original graph built in Phase A. Leaf-branches that are candidates for resolution are highlighted. In this case, the left branch has higher priority and is resolved in this iteration. Figure 5(3) represents the third iteration of the algorithm. The left branch bears now the result of Iteration 1, obtained from the ontology repository (lines 5-7 in Algorithm 1): the ecological ontology states that species *Chromolaena squalida* and *Trichogonia villosa* are preyed on by *Adaina Bipunctata*. By the same token, the middle branch bears the result of Iteration 2 (omitted in the Figure), showing that the geometry for the concept "São Paulo state" is now known. This geometry was retrieved from a geographic Web repository by means of a WFS query execution (lines 8-12 in Algorithm 1).

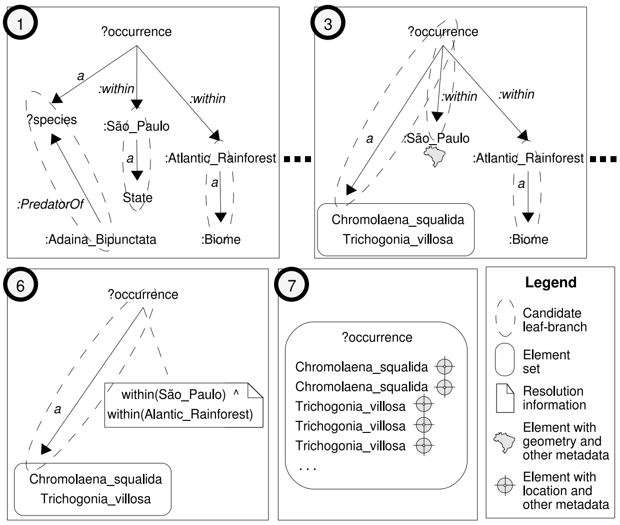


Figure 5. Sequence of states of the query graph for the example query (Figure 4) in successive iterations. Arrows denote the semantics of the predicates and do not imply any orientation to the graph.

Figure 5(6) shows the initial state of the query graph before the last iteration. The graph has been reduced to only one branch. This branch has all information needed to obtain the remote data expressed in the original query: retrieved records must be instances of species *Chromolaena squalida* and *Trichogonia villosa* and must be restricted to the geographic region determined by the intersection of the geometries of São Paulo State and Atlantic Rainforest. With this information, the processor can assemble WFS queries (such as the one shown in Figure 6 - left) and submit them to repositories that, according to the Repository Catalog, may provide the required data (lines 8-12 in Algorithm 1). Figure 5(7) shows the final state of this last iteration. The graph variables are completely resolved, bearing occurrence records of the species requested.

```
<wfs:GetFeature . . . >
 <wfs:Query typeName="plantsorg:species">
  <Filter>
   <And>
    <Or>
     <PropertyIsEqualTo>
      <PropertyName>ScientificN</PropertyName>
      <Literal>Chromolaena_squalida</Literal>
     </PropertyIsEqualTo>
     <PropertyIsEqualTo>
      <PropertyName>ScientificN</PropertyName>
      <Literal>Trichogonia_villosa</Literal>
     </PropertyIsEqualTo>  . . .
    </Or>
    <Within>
     <PropertyName>the_geom</PropertyName>
     <gml:Polygon> . . .
      <gml:coordinates . . . > -46.469289,-18.895586
          -44.87035,-18.66422 . .
    </Within>
   </And>
  </Filter> </wfs:Query> </wfs:GetFeature>
```

```
<wfs:FeatureCollection . . . > . . .
 <gml:featureMember>
  <lis:webios fid=webios.4">
   <lis:the_geom> . . .
    <gml:Point>
      <gml:coordinates . . . >-44.7196,-23.3099 . . .
   <lis:ScientificName>Trichogonia_villosa . . .
   <lis:Collector>A. M. Almeida, U. Kubota . . .
  </lis:webios> </gml:featureMember>
 <gml:featureMember>
  <lis:webios fid=webios.6">
   <lis:the_geom>  . . .
    <gml:Point>
     <gml:coordinates . . . >-44.8341,-23.2024 . . .
   <lis:ScientificName>Chromolaena_squalida . . .
   <lis:Collector>E. P. Anseloni, J.C. Silva . . .
  </lis:webios> . . .
 </gml:featureMember> </wfs:FeatureCollection>
```

Figure 6. (left) Part of a WFS query to retrieve certain species within a given area; (right) GML results for the WFS query containing species occurrence data

The corresponding WFS query (Figure 6 - left) is constructed and sent to the appropriate service. The result is a GML file (Figure 6 - right), corresponding to phase C of the algorithm, and is returned to the query interface.

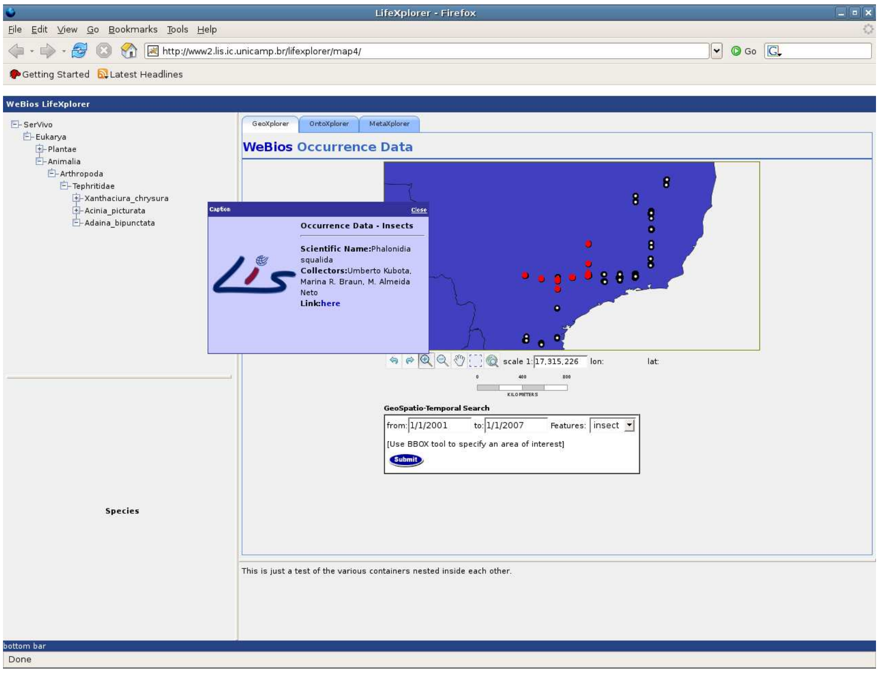
We have implemented parts of a prototype for the query service. We are using

Jena RDF framework [HP Labs] to process (simplified) SPARQL queries and GeoServer WFS implementation [GeoServer Project] to publish repositories.

We have also developed a graphical interface which takes advantage of WFS and WMS services to support user queries. Figure 7 shows a screen copy of this interface. The left part displays a dynamic tree view containing an excerpt of the ecological ontology, which the user can investigate by hierarchical navigation. Points on the map show locations of observations recorded in occurrence records. The window below the map lets end-users define temporal predicates and desired features - in this case, it shows that points display insect information. When the user clicks a point in the map, a query is sent to the species occurrence repositories and returns details on the corresponding record(s). This interface was implemented using Dojo and MapBuilder widget/AJAX toolkits. Dojo is a toolkit that provides richer user interaction and simplifies AJAX programming (it was used, for example, the dynamic tree view). MapBuilder is a toolkit that provides widgets for map interaction. It is responsible for WMS map presentation and WFS query manipulation in the application.

5. Concluding remarks

This paper proposed an architecture for data sharing and retrieval to support biodiversity research. The approach relies on combining information stored in remote data repositories with ecological and geographic ontologies designed by domain experts. Query processing relies on these ontologies, which embed geographic and ecological relations. This extends present biodiversity system mechanisms by supporting a combination of standard spatial and complex ecological predicates.



- SerVivo
  - Eukarya
    - Plantae
    - Animalia
      - Arthropoda
        - Tephritidae
          - Xanthaciura_chrysura
          - Acinia_picturata
          - Adaina_bipunctata

WeBios Occurrence Data

Occurrence Data - Insects

Scientific Name: Phalonidia squalida
Collectors: Umberto Kubota, Marina R. Braun, M. Almeida Neto
Link:here

scale 1: 17,315,226 lon: lat:

0 400 800 KILOMETERS

GeoSpatio-Temporal Search

from: 1/1/2001 to: 1/1/2007 Features: insect

[Use BBOX tool to specify an area of interest]

Submit

This is just a test of the various containers nested inside each other.

Figure 7. Screen copy of the visualization tool using WFS and WMS service implementations

The approach to conciliate the centralized ontological model and the underlying relational data at the repositories contrast with other strategies that aim at deriving ontological models from relational schemas (e.g. [Laborda and Conrad 2006]). We provide a loosely coupled association between domain specific ontologies and repository data. The ontologies and the repositories are independently developed and can be used in other scenarios. This approach simplifies management of distributed repositories and provides higher flexibility to changes in the centralized model; both characteristics are important in the context of biodiversity data sharing.

Though inspired in the biodiversity research domain, we believe that the architecture could be generalized to encompass data in other scientific fields, provided the appropriate ontologies are available. Present work includes defining a comprehensive set of "typical" user queries, together with end users, to test the effectiveness of the proposed framework. Another issue is query performance. Our implementation favors processing via RDF graph management, to take advantage of our ontology structures, and their processing using SPARQL mechanisms. This kind of processing, however, is less efficient, space and time-wise, to process standard predicates. Thus, for large result datasets, a hybrid mechanism is being envisaged, combining SQL and SPARQL.

Acknowledgements: This research was financed by an eScience grant from Microsoft Research, Redmond, and by Brazilian funding agencies CAPES, CNPq and FAPESP.

References

Daconta, M. C., Obrst, L. J., and Smith, K. T. (2003). *The Semantic Web : A Guide to the Future of XML, Web Services, and Knowledge Management.* Wiley.

Elmasri, R. and Navathe, S. B. (1994). *Fundamentals of Database Systems, 2nd Ed.* Benjamin/Cummings.

GeoServer Project. GeoServer web site. http://geoserver.sourceforge.net (Feb 07).

Gomes Jr, L. C. (2007). Uma Arquitetura para Consultas a Repositórios de Biodiversidade na Web (An architecture to query biodiversity data on the Web). Master's thesis, UNICAMP. Supervision C. B. Medeiros.

Gruber, T. (1995). Towards Principles for the Design of Ontologies Used for Knowledge Sharing. *Int Jn of Human-Computer Studies*, 43(5-6):907–928.

Guralnick, R. and Neufeld, D. (2005). Challenges Building Online GIS Services to Support Global Biodiversity Mapping and Analysis: Lessons from the Mountain and Plains Database and Informatics project. *Biodiversity Informatics*, 2:56–69.

HP Labs. Jena website. http://jena.sourceforge.net/ (accessed February 26, 2007).

Laborda, C. P. and Conrad, S. (2006). Bringing Relational Data into the SemanticWeb using SPARQL and Relational.OWL. In Barga, R. S. and Zhou, X., editors, *ICDE Workshops*, page 55.

Medeiros, C. B., Torres, R., Falcao, A., Lewinsohn, T., and Prado, P. (2007). The WeBIOS Project. http://www.lis.ic.unicamp.br/projects/webios (Jun 07).

Morin, P. (1999). *Community Ecology*. Blackwell Science.

OGC (2003). Geography Markup Language (GML) 3.0. https://portal.opengeospatial.org/files/?artifact_id=7174 (accessed February 26, 2007).

OGC (2005a). Filter Encoding Implementation Specification 1.1.0. http://www.opengeospatial.org/standards/filter (accessed February 26, 2007).

OGC (2005b). Web Feature Service (WFS) Implementation Specification. http://portal.opengis.org/files/?artifact_id=8339 (accessed February 26, 2007).

Open Geospatial Consortium Inc. (OGC). OGC website. http://www.opengeospatial.org (Jun 07).

Parr, C., Parafiynyk, A., Sachs, J., Ding, L., Dornbush, S., Finin, T., Wang, D., and Hollander, A. (2006). Integrating ecoinformatics resources on the semantic web. In *WWW '06: Proc 15th international conference on World Wide Web*, pages 1073–1074.

Rodríguez, M. A., Egenhofer, M. J., and Blaser, A. D. (2003). Query Pre-processing of Topological Constraints: Comparing a Composition-Based with Neighborhood-Based Approach. In *Proc SSTD*, volume 2750 of *LNCS*, pages 362–379.

Seaborne, A. and Prud'hommeaux, E. (2006). SPARQL Query Language for RDF. W3C working draft, W3C. http://www.w3.org/TR/2006/WD-rdf-sparql-query-20061004/.

Species Analyst project. Species Analyst website. http://speciesanalyst.net (Feb 07).

Taxonomic Databases Working Group (TDWG). Darwin Core 2 Review. http://darwincore.calacademy.org (Feb 07).

Wilson, E. (1999). *Biological Diversity: The Oldest Human Heritage*. New York State Museum.