A Framework to Process Complex Biodiversity Queries *

Jaudete Daltio, Claudia B. Medeiros, Luiz C. Gomes Jr
Institute of Computing - University of Campinas
CP 6176, 13084-971 Campinas, SP, Brazil
cmbm@ic.unicamp.br

Thomas Michael Lewinsohn
Institute of Biology - University of Campinas
CP 6109, 13083-970 Campinas, SP, Brazil
thomasl@unicamp.br

ABSTRACT

Tackling biodiversity information is essentially a distributed effort. Data handled are inherently heterogeneous, being provided by distinct research groups and using different vocabularies. Queries in biodiversity systems require to correlate these data, using many kinds of knowledge on geographic, biologic and ecological issues. Available biodiversity systems can only cope with part of these queries, and end users must perform several manual tasks to derive the desired correlations, because of semantic mismatches among data sources and lack of appropriate operators. This paper presents a solution based on Web services to meet these challenges. It relies on ontologies to retrieve the query contexts and uses the terms of this context to discover suitable sources in data repositories. This approach is being tested using real data, with new services.

Categories and Subject Descriptors

H.3.5 [**Information Storage and Retrieval**]: Online Information Services—*Web-based Services*; I.2.4 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods—*Semantic networks*

General Terms

Management

Keywords

Biodiversity Systems, Ontologies, Web Services

1. INTRODUCTION

Biodiversity information systems are concerned with the environment and natural resources, to help experts manage information on the various species and the relationships amongst them – e.g., abundance, richness, endemism. This requires managing and correlating species occurrence data with several other kinds of information, such as geographical data (e.g. on habitats or climate variables), gazetteers of toponyms, scientific name checklists, historical records, and many others. Available data are collected all over the world by distinct teams and published in many formats, following a variety of standards. Data volume and species diversity contribute to complicate the issue: estimates for the number of species in the world vary from 10 to more than 100 million [7].

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 groups and institutions. An occurrence record stores data on some kind of observation of living beings – it includes data on a species' taxonomical classification, location where the species were observed or collected, by whom, when and how.

Typical biodiversity information systems are limited in scope, and can only solve a small part of user concerns. Available mechanisms are based on DBMS functions, combining them with spatial correlations. However, biologists also need more complex computations not offered by these systems, such as spatio-temporal correlations or ecological relations among species – e.g., predator-prey relationships. Such relationships must be extracted separately from other sources and deduced by the scientists, who have to invest a considerable amount of time, and execute many manual tasks, to obtain the needed information.

This paper discusses a framework to address this need by combining ontology manipulation (for semantic enhancement) with Web service invocations (for interoperability). Besides supporting the more usual kinds of query predicates, it also allows computation of ecological predicates, by combining stored and derived data and ontological information in distributed data repositories. This solution has been partially implemented using real data within the WeBios eScience biodiversity project [19].

The rest of this paper is organized as follows. Section 2 contains a general description of WeBios and related work. Section 3 presents the framework. Section 4 concerns implementation aspects. A real case study is presented in Section 5. Finally, section 6 concludes the paper.

2. RELATED WORK

2.1 Overview of WeBios

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

SAC'08 March 16-20, 2008, Fortaleza, Ceará, Brazil

This work is being conducted as part of WeBios [19], a biodiversity information system developed within a joint initiative of biodiversity[1] and computer science[2] researchers. Its goal is to provide bio-scientists with a system that supports exploratory queries over heterogeneous and distributed biodiversity data sources on the Web. It has a service-oriented architecture and employs semantic web technologies.

The architecture of WeBios is organized according to four main layers: *Storage*, *Supporting Web Services*, *Enhanced Web Services* and *Client Applications*. The *Storage Layer* is responsible for data storage and low-level data management in distributed repositories, which are fed by distinct biodiversity research projects. There are four kinds of primary data sources: images, species' occurrence records, geographical and ecological data, and ontologies.

The *Supporting Services Layer* comprises five Web services, each of which dedicated to a specific data retrieval modality – images, metadata, geographic data, occurence records and ontologies. The *Enhanced Services* invoke the *Supporting Services* to answer requests that demand combined access to distinct kinds of data sources. *Client Applications* access these services via a mediator, which sends requests to the services and returns the results to the applications. This paper discusses the Ecologically-aware Query – see Section 3– an Enhanced Service of the system.

2.2 Geospatial Services and Standards

Biodiversity data sharing and integration is often based on geographic coordinates. Geospatial Web services and exchange standards for occurrence records are important elements in promoting biodiversity data integration and interoperability among systems [6]. In particular, the Web Feature Service (WFS) [15] specification provides a standardized means to access geospatial data encoded in the Geographic Markup Language (GML) [14]. GML is an XML-based standard for the transport and storage of geospatial information. Another specification WMS (Web Map Service) defines means to produce two-dimensional maps from geospatial data.

There are many initiatives to leverage sharing and interoperability of species occurrence data. Infrastructures for sharing such data on the Internet (such as Species Analyst [3]) rely on exchange standards and transmission protocols to build an interconnected network of data providers. Many solutions adopt Darwin Core [18], an XML-based standard that defines the elements to describe occurrence data.

While geographic services and data exchange standards are important factors in developing biodiversity systems, they solve a small part of heterogeneity issues. They cannot meet user needs concerning, for instance, establishing non-geographic correlations (such as determining food chains or parasitic relationships) or use of multiple user vocabularies. Ontologies are being proposed to support such needs.

2.3 Ontology Servers and Frameworks

From a computer science perspective, an ontology can be viewed as a data model that represents a set of concepts within a domain and the relationships between those concepts. Knowledge in an ontology is formalized using four kinds of components: classes, instances, properties and constraints. Many languages may be used to represent an ontology, such as RDF(*Resource Description Framework*) [13] and OWL(*Web Ontology Language*) [1]. SPARQL [17] is a query language used to query ontologies represented in OWL. There are many ontology tools available, with varying number of functionalities, such as ontology development, merge, annotation, storage, and querying [16].

Several frameworks help the development of applications that need access to ontologies – e.g., Jena [2], SnoBase [9] and SOFA[4]. Usually, such frameworks provide functions to access ontologies that have been stored in distinct formats. In many cases, however, frameworks do not support applications that take ontology evolution into consideration. Indeed, since ontologies describe knowledge about a given domain, they must evolve to reflect knowledge acquisition. Ontology evolution causes considerable application recoding.

Ontology servers have been proposed to solve the need for dynamic management [4, 12]. Similar to frameworks, these servers also support queries to ontologies and, in some cases, also provide reasoners. Some of these servers provide access to ontologies via their URI's, while others store them in a local repository. These servers can only provide access to an ontology at a time, and thus are not appropriate to work in distributed, multi-ontology, scenarios. As will be seen, our solution relies in combining the server and framework approaches.

3. ECOLOGICALLY-AWARE QUERY FRAMEWORK

The Ecologically-aware Query Framework supports queries with complex ecologic predicates, which are evaluated using ontology-based inferences. 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) Web services and standards to access remote data repositories, and (iii) a combination of spatial and ecological predicates to process ecologically-aware queries.

Figure 1 presents a high level view of the framework's architecture [5], and it is composed of two main elements: (i) a query processing module, that processes queries received from Client Applications and (ii) distributed repositories, from where the module retrieves data.

The repositories are databases published by research groups and institutions. There are three types of repositories: for Occurrence Records, for Georeferenced Data (such as lakes, countries or biomes) and Semantic Repositories (containing ontologies and their metadata, managed by an Ontology Service Aondê). All repositories are accessed via Web services.

3.1 Ontological Predicates Module

The ontological predicates module is invoked by the query processor to expand queries and process ecological predicates. To do that, it requests operations from the Aondê Ontology Service [6]. Aondê [3] is part of the Supporting Services of the WeBios (see section 2.1). It manages ontologies that describe taxonomic, ecological and geographic

[1]Insect-Plant Interaction Lab. (LIIP), Inst. of Biology
[2]Lab. of Information Systems (LIS), Inst. of Computing
[3]http://speciesanalyst.net

[4]http://sofa.dev.java.net
[5]Part of the biodiversity project WeBios, described in Section 2.1.
[6]Aondê means "owl" in Tupi, the main branch of native Brazilian languages.

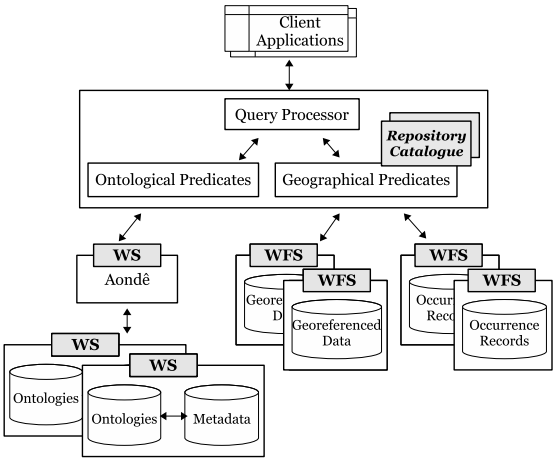

Figure 1: Architecture of the Framework

concepts. This service provides access, management, analysis and integration of ontologies. Ontologies are stored in Semantic Repositories, built and managed by Aondê, organized into ontology and metadata data spaces. Ontology content is provided by research communities. OWL [1] is the language adopted to represent ontologies. The OMV standard[8] is used to represent metadata structures. Aondê allows the following operations:

- **Management of Repositories:** supports insertion, replacement and deletion of ontologies and metadata structures;
- **Search and Ranking:** searches, within a set of repositories, for ontologies that contain a given set of terms. Ranking is based on a set of metrics to analyze the internal structure of each ontology retrieved;
- **Query:** extracts information from an ontology, using SPARQL queries;
- **Difference Detection:** compares two ontologies, considering their structures and contents;
- **View Creation:** constructs a view of a source ontology;
- **Integration:** integrates two source ontologies using the alignment approach, and produces a new ontology.

These operations can be used to find ontologies of interest, to correlate distinct ontologies and build new ontologies to be used in queries by the query processor. Aondê search operation also uses external repository of ontologies, as long as they are available through Web services, and a large biodiversity ontology source, Spire [7], accessible via dedicated calls to the portal.

3.2 Geographical Predicates Module

This module is responsible for performing access to Georeferenced Data and Occurrence Records Repositories. In order to identify which repositories to access, it uses the *Repository Catalogue*, which 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 semantic annotations from ontologies in Semantic Repositories.

[7] http://spire.umbc.edu/ont/ethan.php

Type	URI	Bbox	HasDataAbout
occurrence	http://plants.org/wfs	-46,-18 -43,-16	Chromolaena_squalida, Mikania_purpurascens
occurrence	Http://flies.org/wfs	-47,-12 -42,-15	Tephritidae
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 Catalogue

The *type* field indicates whether the repository contains information on occurrence or geographic phenomena. The bounding box (*Bbox* field) defines the geographic region for which the repository can provide data. The ontologic annotations qualify the contents of a repository. Occurrence data records are assumed to be compliant to the Darwin Core standard [18]. Occurrence and geographic data records are georeferenced (i.e. associated with geographic coordinates) and must be compliant with the WFS service standard, thus standardizing interfaces and providing means to apply geographic filters in data retrieval.

3.3 Query Processor

The query processor is responsible for coordinating the processing of a query, being schematically illustrated in Figure 3. It receives as input a set of query expressions, which corresponds to the translation of a client application request, containing complex ecologic predicates. The result of the query processor is a GML file containg the answer to the request. The three main processing phases are:

A) Disambiguate Semantics: domain-specific terms in the input query are disambiguated by repeated invocations to the Aondê service. The result of this stage is an XML file, result of a SPARQL query, that is sent to the next module;

B) Get Georeferenced Data: this module starts by querying the Repository Catalogue to determine possible data sources (matching query terms with ontologic annotations and checking the bounding boxes). Next, it sends WFS requests to these repositories, retrieving geographic data and occurrence records. The result is a GML file.

C) Merge Results: phases (A) and (B) are executed for each query expression of the input, and this phase merges the GML results.

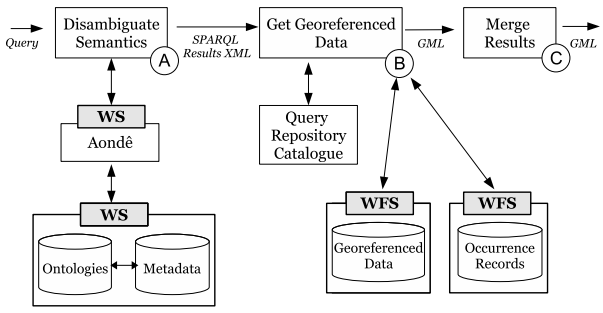

Figure 3: Query processing phases

4. IMPLEMENTATION ASPECTS

All repositories have been implemented, using real data, stored in the PostGIS object-relational database system. A prototype of Aondê has been implemented in the Java language. Access and navigation over ontology contents are provided by the Jena framework [2]. This version of Jena

is composed by an RDF API, an OWL API, in-memory and persistent storage (in relational databases), a SPARQL query engine and a rule-based inference engine.

The Aondê Web service implementation uses Apache Axis, an open source Web service framework. It consists of a Java implementation of the SOAP server, and various utilities and APIs for generating and deploying Web service applications. Customized Web service deployment requires a specific descriptor called WSDD (Web Service Deployment Descriptor), used to specify resources that should be exposed as web services.

Georeferenced and occurrence repositories are published in the WFS standard through GeoServer – see Figure 4. WFS' methods *GetCapabilities* and *DescribeFeatureType* are first used to retrieve the structure of data in each repository. Next, this structure is used to construct *WFS GetFeature* requests to retrieve the desired records. *GetFeature* invocations contain *Field* clauses that specify spatial and standard predicates (e.g., species names or timestamp).

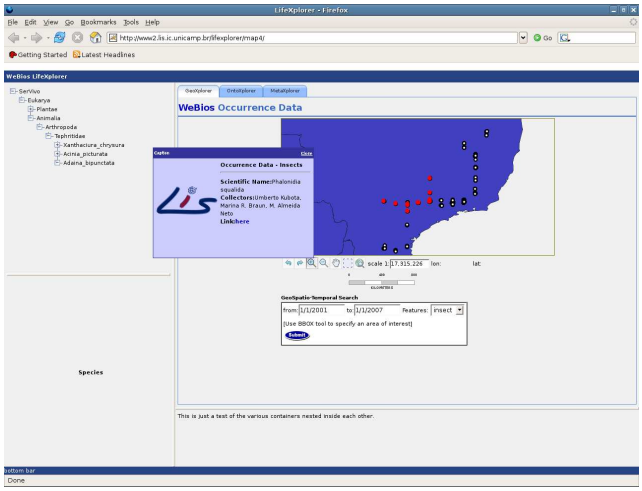

Figure 4: Query result: visualization of GML data

5. CASE STUDY

The case study stems from a long-term ecological project, concerning a large array of plants and the insects that feed on their reproductive structures. The study focuses on a particular family of plants, the *Asteraceae* or *Compositae*. This is the largest family of flowering plants worldwide, with 25.000 known species; these include many edible plants such as sunflowers and lettuce; ornamentals, such as dahlias and chrysanthemums; and serious pest weeds, such as thistles. The insects that feed on their flowerheads are also highly diversified, including many species of flies, moths, and beetles.

WeBios biology experts collected data over eight years to explore assemblages of plants and the insects associated with them across various spatial scales and in different biogeographical and ecological settings. The large array of plants and insect records obtained of field work allows to explore such questions as: *"how constant and predictable are local insect-plant interactions among similar localities?"*, *"does the variation in species composition or interactions show evidence of spatial autocorrelation?"* and *"how do these species arrays and their interactions change over different spatial and ecological scales?"*.

Such questions cannot be directly answered by existing biodiversity systems, and open up many new possibilities of investigating biodiversity issues that are not feasible with the usual data and analyses [10, 11]. Such advances are needed to improve the quality and range of predictions demanded for biodiversity monitoring, conservation, management and sustainable use.

Example 1

We built a large ontology (over 2000 terms) reflecting domain semantics, here called Eco-Onto. An example of a typical query in the context is *"For occurrence records of year 2000, what were the insect species most frequently collected in the flowerheads of the Trixis verbasciformis plant, in Cerrado* [8] *areas"*. This query is decomposed into two distinct queries: (a) determination of insects that are found in the plant; and (b) determination of the geographic extents of Cerrado. The result is the intersection of (a) and (b) and is processed in our framework as follows:

1. Process query (a)

1(a). Module **Disambiguate Semantics** invokes the Aondê service. This invocation is a SOAP message requesting a query operation, to be posed to Eco-Onto. From a high level point of view, this invocation has the form "QUERY (ECO-ONTO, SPARQL, QUERY-STRING)", which indicates that a QUERY operation is requested to Eco-Onto, that it is expressed in SPARQL and is stated in the QUERY-STRING parameter. In particular, this query requests *"The names of insect species most frequently collected in flowerheads of species Trixis verbasciformis"*. The service returns the answer to this request in an XML file, embedded in a SOAP message. This file is forwarded to step **2(a)**.

2(a). Module **Get Georeferenced Data** extracts species names from the XML file, and checks, using annotations of the Repository Catalogue, which Occurrence Repositories may contain occurrence records of these species. The module sends to each such repository a (WFS) request for occurrence records, passing species names and "*year = 2000*" in *Filter* fields. The results of these WFS requests are GML files, containing the occurrence records (e.g., who, when, where and how the species were collected). GML files are passed on to the **Merge Results** module.

2. Process query (b)

1(b). Disambiguation of term Cerrado requires a new request to Aondê, to pose a SPARQL query to another ontology – Biome ontology[9]– requesting all ecoregion names associated with Cerrado. This request is answered by a message containing an XML file, forwarded to step 2(b).

2(b). Module **Get Georeferenced Data** extracts ecoregion names from the XML file and, after checking for relevant repositories in the Repository Catalogue, sends (WFS) requests to Georeferenced Data Repositories. The result is a set of polygons in GML that delimit the regions of interest.

3. Merge results

Module **Merge Results** finds the intersection of the GML files obtained in steps **1(b)** and **2(b)**, selecting occurrence record whose coordinates fall within some Cerrado polygon.

We point out that another possibility to process this query would be to start by query (b) obtaining Cerrado polygons

[8]Brazilian savannah

[9]A biome is an ecological community type – e.g., rainforest, savannah.

via *GetFeature* commands, and then proceed to query (a) to obtain occurrence records that fall within these polygons. Query processing strategies require performance considerations, which are outside the scope of this paper.

Example 2

A more complex (and common) scenario is the one where no single ecological ontology contains all data needed to satisfy a query, and biologists need to combine information from two distinct collection ontologies – Eco-Onto and Col-Onto (defined by another group), for instance. Moreover, the Eco-Onto ontology contains some insects whose species have not been completly identified. This new query might be *"For occurrence records of year 2000, what were the unidentified insect species most frequently collected in the flowerheads of the Trixis verbasciformis plant, in the Cerrado"*. This query can be processed akin to the previous example; however, it will need a few preliminary invocations to Aondê, to create an ontology that will serve as input to step (**1(a)**). A possible invocation sequence would be:

1. Request a view from Eco-Onto, containing all unidentified insect species – in high level, a SOAP message containing VIEW(UNIDENTIFIED, PREYEDON:1, SUBCLASSE:2, HASSPECIES:1, INSTANCE);

2. Extract the same kind of view from Col-Onto;

3. Integrate both views, using an invocation of Aondê requesting execution of the INTEGRATION operation on the two views. This operation will align terms from both views, defining equivalences among concepts used by the two research groups involved. The result of this integration will be the input ontology to step (**1(a)**).

6. CONCLUSIONS

This paper discussed a query processing framework to support biodiversity research. The approach relies on combining information stored in remote data repositories with ecological and geographic ontologies designed by domain experts, embeding geographic and ecological relations. This extends present biodiversity system mechanisms by supporting complex ecological predicates and multi-ontology management. Our solution has been implemented using real data and case studies.

While ontologies enhance semantics and allow computing new kinds of predicates, Web services and standards support interoperability across distinct tools and repositories published by distributed research groups. Present work involves many issues. We are developing basic client applications to provide adequate end user interfaces. Another issue is query performance. Our implementation favors query processing on RDF graphs and SPARQL mechanisms to take advantage of our ontology structures. This kind of processing, however, is inadequate to process standard predicates. Thus, for large result datasets, a hybrid mechanism is being envisaged, combining SQL and SPARQL. For more details on these and other extensions, the reader is referred to [3, 5, 19].

7. REFERENCES

[1] G. Antoniou and F. van Harmelen. Web Ontology Language: OWL. In S. Staab and R. Studer, editors, *Handbook on Ontologies in Information Systems*, pages 76–92, 2003.

[2] J. Carroll, I. Dickinson, C. Dollin, D. Reynolds, A. Seaborne, and K. Wilkinson. Jena: implementing the semantic web recommendations. In *Proc. of the 13th international WWW*, pages 74–83, 2004.

[3] J. Daltio and C. B. Medeiros. An ontology service for biodiversity information systems (in portuguese). In *XXXIV SEMISH: Seminar on Software and Hardware*, pages 2143–2157, Rio de Janeiro, Brazil, July 2007.

[4] M. Duke and M. Patel. An Ontology Server for Agentcities.NET. Agentcities Task Force Technical Note, September 2003.

[5] L. C. Gomes Jr. An architecture to query biodiversity data on the Web (in portuguese). Master's thesis, State University of Campinas - UNICAMP, May 2007.

[6] R. Guralnick and D. Neufeld. 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, Aug. 08 2005.

[7] P. Hammond, B. Aguirre-Hudson, M. Dadd, B. Groombridge, J. Hodges, M. Jenkins, M. Mengesha, and W. S. Grant. The current magnitude of biodiversity. Global biodiversity assessment, 1995.

[8] J. Hartmann, Y. Sure, P. Haase, R. Palma, and M. C. Suárez-Figueroa. OMV – Ontology Metadata Vocabulary. In *ISWC 2005 - In Ontology Patterns for the Semantic Web*, November 2005.

[9] J. Lee. An Application Programming Interface for Ontology. IBM T. J. Watson Research Center. Document from SNOBASE v.1.0 release documentation, November 2003.

[10] T. Lewinsohn, P. Prado, P. Jordano, J. Bascompte, and J. Olesen. Structure in plant-animal interaction assemblages. Oikos 113174-184, 2006.

[11] T. M. Lewinsohn and G. J. Shepherd. Taxonomic knowledge bases as tools for biodiversity research in the third world. In *Proc. VI International Congress of Ecology*, page 77, Manchester, England, 1994.

[12] Y. Li, S. G. Thompson, Z. Tan, N. Giles, and H. Gharib. Beyond Ontology Construction; Ontology Services as Online Knowledge Sharing Communities. In *International Semantic Web Conference - ISWC 2003*, volume 2870 of *LNCS*, pages 469–483, 2003.

[13] F. Manola and E. Miller. Resource Description Framework (RDF) Model and Syntax Specification, February 2004. http://www.w3.org/TR/rdf-primer/.

[14] OGC. Geography Markup Language (GML) 3.0. https://portal.opengeospatial.org/, December 2003.

[15] OGC. Web Feature Service (WFS) Implementation Specification. http://portal.opengis.org/, May 2005.

[16] A. G. Perez, J. Angele, M. F. Lopez, V. Christophides, A. Stutt, and Y. Sure. A survey on ontology tools. Deliverable 1.3, EU IST Project IST-2000-29243 OntoWeb, 2002.

[17] E. Prud'hommeaux and A. Seaborne. SPARQL Query Language for RDF. Technical report, World Wide Web Consortium - W3C, 2006.

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

[19] WeBios. Web Service Multimodal Tools for Strategic Biodiversity Research, Assessment and Monitoring. http://www.lis.ic.unicamp.br/projects/webios, 2007.