BioGeoSDI workshop - GeoInteroperability Testbed Pilot Project

BioGeoSDI workshop - Version 0.6

TDWG Access Protocol for Information Retrieval (TAPIR) Specification

Version: 0.6

Date: 05 July 2007

Documentation Editors

- Javier de la Torre (jatorre [at] imaste-ips [dot] com)
- Tim Sutton (tim [at] linfiniti [dot] com)
- Bart Meganck (bart.meganck [at] africamuseum [dot] be)
- Dave Vieglais (vieglais [at] ku [dot] edu)
- Aimee Stewart (astewart [at] ku [dot] edu)
- Peter Brewer (p.w.brewer [at] reading [dot] ac [dot] uk)

Document History

- Initial template version: April 1st, 2007 (during the Campinas meeting)

---

1. Abstract

A week long workshop was held in Campinas, Brazil during the first week of April 2007. The focus of the workshop was to develop a testbed web application that demonstrates the interoperability of digital data and services using open standards - with particular emphasis on geospatial, taxonomic and occurrence biodiversity data.

Two prototype web applications were developed in php and Flex. The wizard style application leads the user through a defined sequence of steps in order to acquire sufficient data to create a niche model. The process includes taxonomic validation using the Catalogue of Life, search and retrieval of occurrence data using services such as GBIF portal or WFS, selection of raster layers representing environmental data needed in the modeling process, and modeling these data using the openModeller Web Service in order to create a probability surface that represents areas where a species is likely to be able to survive.

The workshop highlighted just how easy it is to rapidly create such a feature rich application using open access to data, free software and open standards. It also highlighted some areas where further work is needed in order to truly be able to blend these kinds of services into a cohesive computing platform. Finally, suggestions were made for improving OGC standards and data quality.

2. Introduction

Biodiversity informatics is a rapidly growing field in which digital data is increasingly available. Taxonomic data, species occurrence records, ecological data, and environmental data are all available online through various services provided by scientists worldwide. This data is published in different formats and protocols, which make them difficult to be used as a single service to complete a real work.

3. Background

At the heart of biodiversity informatics there is the dream of a unified infrastructure where data and analysis services all around the world are seamlessly used.

A common goal of TDWG participants is an environment where different TDWG initiatives interoperate to provide rich mechanisms for biodiversity knowledge exploration, analysis, and discovery. However, though there is a general awareness of relationships between TDWG standards and groups, there are few examples of practical inter-standard applications.

At the same time most of biodiversity research has to deal with geographical information, biodiversity happens somewhere. In the geospatial world there is an equivalent organization to TDWG called OGC which promotes interoperability among geospatial services. They accomplish this by creating a set of open standards that different vendors and projects are using to implement their geospatial infrastructure.

We examined, by way of example software applications, the degree of interoperability that can be achieved by TDWG, OGC, and related initiatives (proposed and existing) such as the GBIF REST occurrence service, openModeller web services (OMWS), TAPIR and LSIDs.

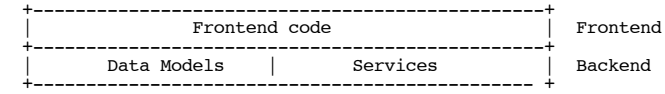
To achieve these objectives, a small group of developers with expertise in the relevant domains held a workshop to develop and example web application. This application retrieves data from specimen and observation sources (GBIF Cache, TAPIR or WFS) based on scientific names, with synonym resolution. Using environmental data accessible through OGC services, the application generates ecological niche models through openModeller web services (OMWS). The result can in turn be accessible as OGC WMS and WCS layers. While this is not a novel application, previous experience indicated that there are many issues with data access, quality, processing, and standards interoperability that limit the generalized implementation of such an analysis pipeline.

The outcome of the workshop is an example application that binds these core standards and data sources. More important though, is the identification of problem areas within the existing standards and the recommendations on suggested improvements to the respective TDWG or other groups. Though TDWG has developed several key interoperability standards, it has not invested sufficiently in the practical application of these standards.

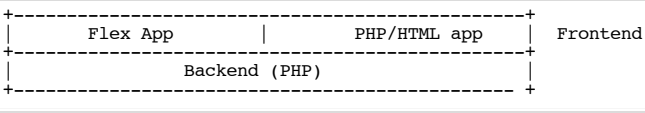
4. Architecture Overview

The architecture was designed in order to meet the objectives of: 1. rapid application development 2. clean separation of presentation and logic layers 3. the easy addition of new services to the framework

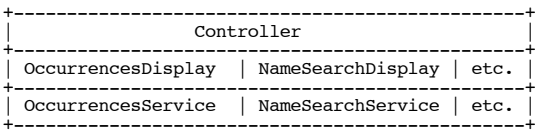
In order to achieve this a tiered architecture was created consisting of Frontend and Backend code, with the backend code arranged into DataModels and Services:



Two user interfaces were developed simultaneously - a 'low entry bar' PHP/HTML interface and a 'high entry bar' PHP/Flex interface. The Flex interface requires Flash 9 to be available in the client browser. The results of the computations from the various services are stored in data models and passed back to the display layer to render them to the screen.



The PHP/HTML Frontend code was separated into a Controller and various Display classes :



5. Implementation

5.1. Name Search service

The name resolution service takes as input a (part of an) scientific name and returns a list of matching names. Together with every name that is returned from the service the correspondent accepted name is also returned. This service forces the user to choose a single accepted name, either directly or trough one of its synonyms.

5.2. Occurrence Search service

The Occurrence Search service takes the scientific name obtained from the name search service, and returns a list of species occurrences (specimen or observations) with latitude and longitude. The user can select various occurrence providers in the interface. Three different technologies were implemented, which together give a wide scope of possibilities :

- GBIF REST occurrence service. Not a real standard but a popular source of occurrences.
- a WFS service that implements a GML application schema that the prototype can understand.
- a TAPIR provider whose data has been configured to allow the use of DarwinCore 1.4 or ABCD v2.06

The biggest challenge here was not the use of the different transport protocols but the semantic mediation needed to use data of the same type (occurrences), described in many different ways: our own OGC GML Application schema, GBIF XML format, ABCD and Darwin Core. Especially problematic is the use of WFS to distribute occurrence data. There is no official or unofficial GML Application schema to distribute specimen occurrence data and therefore every service could be creating its own (like we did for this experiment) and interoperability will not be possible.

Technically it was easier to consume data from GBIF and TAPIR providers as it is possible to access them directly using simple REST queries, URL + parameters. In the case of WFS the queries were also done using a REST query but the filter on it was actually an URL-encoded piece of XML. This makes a very ugly URL, difficult to debug with a simple web browser.

Behind the scenes the prototype is doing several things after retrieving data from these different sources: 1/ Gathering the data 2/ Send it to a data processing service which: 1/ Inserts the data in a new table in a PostGIS database 2/ Registers the newly created dataset in Geoserver as a WFS FeatureType and a WMS layer.

Once the data is retrieved and cached in the PostGIS database, it becomes available locally as a WMS, WFS, WCS service, KML, PDF and other formats thanks to Geoserver.

5.3. Environmental Data Layer Selection

This service provides the user with a choice of available environmental data layers for use in the modelling calculations. Two different sources can provide these layers :

1/ The openModeller cache of environmental layers. As part of the openModeller Web Service, the getLayers call returns a simple XML collection of layers that are available on the modelling server.

2/ an OGC WCS server. The prototype managed to get a list of a available layers in a certain WCS server, but failed to retrieve them, due to lack of time.

The process is more complicated in this prototype. After retrieving the layer from WCS, it has to be pushed to the modelling service to make it available for later processing. It was not possible at this moment to consider Open Modeller to use remote layers in WCS servers, but this could become easier now that the GDAL library, which OpenModeller uses, has support for WCS.

The use of WCS services was only partly implemented as we had problems with the WCS server we were using for testing. It does not look too difficult to complete support for WCS in the prototype.

5.4. Niche Modeling service

To provide niche modelling capability in the prototype, the openModeller Web Service (OMWS) interface was used. openModeller is a generic framework for carrying out fundamental niche modelling experiments - typically used to predict species distribution given a set of environmental raster layers and a set of known occurrence points. The OMWS interface provides access to this library using SOAP (Document / literal style). Besides calls directly related to modelling, the OMWS provides additional functions for retrieving metadata and other information. See for example the getLayers call used in the Environmental Data Layer Selection tool.

Before going to modeling, the prototype shows a list of available algorithms in the OWS service, for the user to choose from. This is done with a method called getAlgorithms.

The niche modelling step takes in the environmental data layer(s) and the occurrence data, runs a niche modelling experiment, and returns a model and a raster layer representing the probability surface for the model. The OMWS saves the resulting raster layer and creates a WCS service to return the data to the calling server.

Eight algorithms are provided by OMWS, that normally are configurable by the user. For the prototype however, default parameters are pre-set by the application. The user does not need to make any decisions other than selecting an algorithm.

6. Conclusions

The exercise allowed us to divide existing standards in three categories, based on how easy their implementation and use was in our demonstrator application :

1/ Standards where modification or extension would be welcome.

2/ Standards that the community has supported in theory, but that have proven problematic because of lack of existing implementations. The lack of implementations probably hints to underlying problems with their ease of use and with the true commitment of the community.

3/ Standards that excel in terms of community support, existing implementations, and ease of use.

This is, of course, a rude simplification. None of the standards really fits into a single one of these groups. But it was useful to focus the mind, so that an overview could be drafted of the priorities that our community should consider when supporting a standard. With this in mind, we'll describe each of the used standards in a bit more detail now :

6.1. Life Science IDs (LSID)

Though LSIDs have been adopted by TDWG as a community standard, there are few projects that have implemented them. It would greatly improve interoperability to have data objects tagged with LSIDs but their implementation is difficult for many institutions. In any case we could not find any service offering LSIDs relevant to the experiment and together with the lack of expertise within the group, finally LSIDs were not used.

6.2. OCG Webservices

We achieved significant success using Open Geospatial Consortium (OGC) standards WMS, WFS, and WCS. All these OGC web services specify a GetCapabilities operation which returns a description of the data available from an OGC service provider : this is very useful to get a quick overview (e.g. to list the available layers for the user to choose). The specifications are also very clear on the syntax of the key-value pairs to be encoded on the URL GET string for accessing the data. Accessing the documents was easy, but there is a lack of simpler examples on how to use the different standards. The specifications are too long and detailed for potential hackers to get into them.

6.2.1. WMS (Web Mapping Service)

WMS is a simple web service that returns a map view of spatial information as an image. The BioGeoSDI experiment uses the WMS standard to display occurrence data, environmental data, and completed niche models. Because of the nature of WMS, every request to the server has to be dynamically created and therefore the use of WMS can be very resource intensive for the server. Considering the use of WMS together with services like Google Maps, we had the possibility to do some caching to avoid recreating dynamically the maps, but this is maybe something that could be tackled by OGC.

6.2.2. WFS (Web Feature Service)

WFS is a service that returns vector data, geographical features, in XML format. WFS requires data providers to encode their data in a domain-specific 'Application Schema' which references GML (Geographic Markup Language) for primitive geometry object types. The BioGeoSDI experiment used WFS to obtain specimen occurrence data using an specific GML application schema we created. See GML and GML application schemas for more details.

Additionally, even if the prototype wouldn't have make use of it, we found a missing functionality in WFS, paging. Considering that the kind of experiments biodiversity needs to make it seams unfeasible to retrieve all features at once from a WFS service to do an analysis with them. Because of the lack of paging, if a service has set a limit of 1000 features per request, analysis tools would never be able to get further than that in an standard way. Tricks like requesting per boxes could be done it could easily be that a service is exposing several thousand features with the same coordinates and therefore this "paging mechanism" do not work. Would be desirable to have an standard paging mechanism in WFS.

6.2.3. WCS (Web Coverage Service)

WCS is a service which specifies a GetCoverage operation that returns raster data. The BioGeoSDI experiment used WCS providers to obtain environmental data layers.

6.2.4. GML (Geographic Markup Language)

GML is the XML format used by OGC services as payload in services that produce XML, as WFS. GML provide the basis for creating XML formats that have geographical data. We use it in our experiment together with WFS. Read the next point on GML application schemas.

6.2.5. GML application schemas

The application schema is a standard for exchanging data within a community. We could not find any GML application schema that could be used for sharing occurrence data so we created our own GML application schema and set up a sample WFS server. We knew there is a GML application schema implementing the Darwin Core elements coming, but at this time it is not ready. We found not easy in any case to make use of any arbitrary schema due to the lack of support for complex GML schemas in Open Source WFS implementations. At the end, even if we created support for GML in the prototype it would not be very useful as there is no provider implementing the GML application schema we created and understand. So the use of GML alone does provide a very interoperability benefit for biodiversity. Without a community GML application schema that providers and clients use there is no way to use data from different services together.

6.3. Communication/integration protocols from biodiversity world

6.3.1. TAPIR

TAPIR is an acronym for TDWG Access Protocol for Information Retrieval. It was envisaged as a protocol for unifying existing biodiversity data sharing networks based on DiGIR and BioCASe. The protocol can be well compared to WFS, but it has no binding to any XML schema, like GML in WFS, as it could be used with any XML schema.

Due to the use of the protocol only within biodiversity almost all services have their data mapped to ABCD and Darwin Core concepts and therefore it is to understand what TAPIR providers produce as response from a client point of view. The GBIF portal can index also TAPIR providers with Darwin Core and ABCD.

Our experiment successfully implements a TAPIR client for accessing specimen occurrences. Of course, like with WFS, we could only make use of TAPIR servers that have their data mapped to Darwin Core or ABCD 2.06 concepts.

Due to the popularity of the TAPIR protocol within the biodiversity community is easier to find services that the prototype can access, but this was because of the knowledge within the group, not because of the information available on the web. Specially we could not find any TAPIR provider registry.

6.3.2. SPICE

The SPICE protocol was one of two Catalogue of Life products used for taxonomic verification purposes within the BioGeoSDI experiment. Both systems enable the user to provide a potential ambiguous taxon string and verify it's taxonomic status and currently accepted name.

The SPICE (Species 2000 Interoperability Coordination Environment) protocol was developed by researchers at the Universities of Cardiff, Reading and Southampton for use by the SPICE software to query distributed global species databases as part of the Catalogue of Life Dynamic Checklist. The SPICE protocol was therefore originally designed for communications between global species databases and the central SPICE server rather than as a publicly available web service.

The BioGeoSDI experiment makes use of the SPICE wrapper to the Catalogue of Life Annual Checklist. The Annual Checklist edition of the Catalogue of Life is manually compiled from database exports provided by partner database custodians. The SPICE wrapper to the Annual Checklist was written by the Species 2000 Secretariat as a step towards unifying the Dynamic and Annual Checklists into a single system.

A system for taxonomic verification was successfully implemented in the BioGeoSDI experiment using the SPICE protocol. Users can use SPICE to obtain confirmation on the currently accepted taxon name before continuing further with their modelling experiment. The SPICE protocol is capable of providing full synonym for a taxon (i.e. all known synonyms for a species) which could potentially be used to query data stored under

any known name from other databases. Unfortunately time pressures during the BioGeoSDI workshop meant that this functionality could not be explored. Instead the taxonomic verification step provided by SPICE confirms the status of the name provided and returns the currently accepted name if the species is a synonym.

Whilst the SPICE protocol includes all the data required for taxonomic verification, the XML schema was a little cumbersome to use. The nested nature of the infraspecific taxon information and the difference in tag of homologous data in synonyms and accepted names led to code that was more complicated seemed necessary. From a programmers viewpoint the use of capitalised tag names throughout the schema made for uncomfortable programming and the switch to CamelCase in a number of tags is frustrating.

One data problem that arose through using SPICE was the lack of provenance data associated with the taxonomic records. As the SPICE protocol was designed for communications with a specific GSD (Global Species Database) and a centralised SPICE server, database metadata is provided on a per wrapper basis. This means that the correct level of citation of data as required by the Catalogue of Life end user licence cannot be provided.

6.3.3. Catalogue of Life Annual Checklist Web Service

The Catalog of Life (CoL) project is a joint effort between ITIS and Species 2000. CoL provides an Annual Checklist of taxonomic names distributed in CD-ROM and avaialble as a Web Service.

The Catalogue of Life Annual Checklist Web Service was written in response to a call from the users to have programmatic access to the Catalogue of Life. The prototype was released concurrently with the BioGeoSDI workshop and therefore support for this web service was only completed after the workshop was completed.

This web service provides all the data required by the BioGeoSDI experiment for the taxonomic verification purposed including the provenance data missing from the SPICE protocol. The schema provides easier access to individual data elements than SPICE, requiring approximately one third the lines of code to extract the same data.

6.4. Contributing Community Projects

The next set of services does not represent any actual standard, but because of their wide use within the community we consider interesting describing them here and use them in the workshop.

6.4.1. GBIF REST services

The Global Biodiversity Information Facility (GBIF) REST services provides access to records of the occurrence of records from the GBIF cache of community records from hundreds of providers. By caching the data from lot of different biodiversity providers it is a very convenient way of abstracting the complexity of consuming data from several providers at once. The GBIF services are created following the RESTful paradigm. Accessing RESTful services was the easiest, but the functionality and the quality of the service was a little bit problematic. There are a lot of occurrences exposed through the service that does not have coordinates which are useless for this experiment and there is no way to filter them when accesing the service. Since the prototype the services has been upgraded and this problems does not happen anymore.

The documentation available for the services is good and fast to read. To get to it you need to call the services without parameters. Maybe it would be interesting to have a page somewhere more accessible where developers can find this documentation, for example a link on the GBIF website menu for developers.

6.4.2. openModeller Web Service (OMWS)

The openModeller project has provided the OMWS for creating ecological niche modelling experiments. The openModeller team has created a very simple API with few methods. The idea is that different niche modelling programs implement the API and therefore standard clients, accessing through web services, can use seamlessly all them. Up to now only the openModeller team has created a server that implements the OMWS, but more hopefully will appear on the future. The web service is created using SOAP Document/Literal. It turned to be complicated to use SOAP document/literal in PHP because of the lack of support for this kind of web service. We used an early version of a library called NUSOAP that provides SOAP client functionality, but finally we had to unmarshal the messages comming from openModeller. So it turned that using SOAP was a problem on interoperability on this case. Specially considering that there is no direct benefit of using SOAP over other kind of web services type, at the end you have to take care of parsing the messages your own, it

might had been easier if OMWS would be using a different type of web service.

Just as a note here, it might be possible to implement the OMWS using the OGC Web Processing Service (WPS), but still nothing has been studied on this direction.

7. Recommendations

7.1. More clarity in OGC standards publications

OGC WMS, WFS and WCS standards are well-defined and easy to use, with very useful functionality (like the GetCapabilities call). Sadly, the standards publications are somewhat confusing and difficult to navigate. Their usability would be greatly improved if they were published :

- in easily browseable web pages
- with good search capacities
- with many real-life examples
- with many hyperlinks

7.2. A warm call for better data quality

The amount of online data is stunning, spelling a great future for integrated standards-based webtools. But data quality is often poor. This is a serious issue, as online data is not just some "nice to have" publicity front on the web. More and more, it will become the raw data for serious scientifical research (statistics, modelling), leading to publications. This cannot become reality unless the utmost care is given to filling all data fields (coordinates !), and providing an accuracy estimation where appropriate. Some of the problems we encountered were :

- many species are available in one data provider, but not in others
- GBIF data does not reference the original data providers
- Occurrence data quality is often poor (many blank data fields, such as coordinates)
- latitude/longitude values have no accuracy indication, including lack of SRS info and that is definitely a big source of inaccuracy.

These issues will seriously hamper any development of integrated standards-based webtools such as our demo application.

7.2.1. DarwinCore

Currently, DiGIR providers frequently use DarwinCore. But this schema is being served in 24 unique variations, with five defined as only of 'historical significance' (so all the others are still in active use). Only a single version (1.3) is officially accepted by TDWG, and only a single data provider (Cornell) uses it.

In its basic form, DarwinCore is quite minimalistic, only requiring a small set of (meta)data fields. For geospatial (GIS) applications, geographic extensions have been defined, but they are not official standards yet. add text about suitability for GML application

7.2.2. ABCD

The latest official schema, approved as a TDWG standard, is 2.06. There are several provider already using it. Points to edit: -There are lot of different ways to represent geospatial information, but most providers make use of corrdinates. -Complicate to parse due to its principle of variable atomization. The same information is provided in different ways and the clients making use of it has to deal with the merging the data from different formats and levels of atomization.

8. Technologies used on this experiment

8.1. Programming languages

- PHP
- ActionScript
- Bash shell scripting
- Python

8.2. Open Source projects

- Geoserver: WFS, WMS server. KML and PDF creation.
- PostGIS: Temporary geospatial datasets store
- MapServer: WCS
- OpenModeller: OMWS

8.3. Term Definitions

ABCD	Access to Biological Collection Data. See http://www.bgbm.org/TDWG/CODATA/Schema/default.htm
BioCASe	Biological Collections Access Service. See http://www.biocase.org
Catalogue of Life	-
DarwinCore	-
DiGIR	Distributed Generic Information Retrieval. See http://www.digir.net
Flex	-
GBIF	Global Biodiversity Information Facility. see http://www.gbif.org
GBIF portal	see http://data.gbif.org
GDAL	Geospatial Data Abstraction Library. see http://www.gdal.org/
Geoserver	-
GML	Geography Markup Language. see http://www.opengeospatial.org/standards/gml
GSD	-
KML	see http://code.google.com/apis/kml/documentation/
LSID	Life Sciences Identifier. see http://lsid.sourceforge.net/
OGC	Open GIS Consortium. See http://www.opengeospatial.org
OMWS	-
openModeller	-
php	-
PostGIS	-
REST	Representational State Transfer. see http://www.xfront.com/REST-Web-Services.html
SOAP	Simple Object Access Protocol, an XML-based messaging protocol used for invoking web services and exchanging structured data.
SPICE protocol	-
TAPIR	TDWG Access Protocol for Information Retrieval. see http://www.tdwg.org/dav/subgroups/tapir/1.0/docs/TAPIRSpecification_2007-02-24.html
TDWG	Taxonomic Databases Working Group. See http://www.tdwg.org
WCS	Web Coverage Service. see http://www.opengeospatial.org/standards/wcs
WFS	Web Feature Service. see http://www.opengeospatial.org/standards/wfs
WMS	Web Map Service. see http://www.opengeospatial.org/standards/wms
WPS	Web Processing Service. see http://www.opengeospatial.org/standards/requests/28