Annex 1
The DaRWIN Collection Management System

1. INTRODUCTION **3**

2. METHODOLOGY **3**

3. INFRASTRUCTURE **7**

3.1 Technical update of DaRWIN core components 7

4. RESULTS AND RECOMMENDATIONS **9**

4.1 Renormalization of localities and collecting dates 9

4.2 Enhanced geographical information 12

4.2.1 Integration of geographical OGC Web Map server 12

4.2.2 Integration of free-hands polygons and lines in the query interface 17

4.2.3 Satellite background layers 18

4.2.4 Translation of geographical names 18

4.2.5 Automatic conversion DMS/ DD coordinates 19

4.3 Parallel taxonomies 19

4.4 Integration of stable identifiers 20

4.5 Reworked asynchronous mechanism for reports in tab-delimited format 23

4.6 Export to DaRWINCore Archives 24

4.7 Automatic unique collection numbers and uniqueness check 24

4.8 Distinction between taxonomic hierarchy and determination events 25

4.9 Rotating backup 26

4.10 Embedded PL\PgSQL REST client 27

4.11 Introduction of identifiers for linked data 29

4.12 Autocomplete fields 29

4.13 Mass actions 29

4.14 Better Navigability 29

4.15 Count of specimens 30

4.16 New Import system using several XLS templates 30

4.17 New module for labels creation, export and printing 31

4.18 New module for the loans. 31

5. DISSEMINATION AND VALORISATION 32

6. PUBLICATIONS 33

7. ACKNOWLEDGEMENTS 33

1. INTRODUCTION

The inception of the DaRWIN system at RBINS dates back from the 2009-2010 period, and its developments took place between 2011 and 2016. As a matter of fact, DaRWIN is the reworking of an anterior collection management system (“DaRWIN I”), used at the RBINS, that had been developed in Oracle and used a custom-made desktop interface in Java. For the second iteration of the project, the development team led by Paul-André Duschesnes decided to adopt Open-Source technologies and to offer a web-based solution that could work both on the Intranet and Internet.

The second version of the DaRWIN databases would then adopt the following technologies:

- PostgreSQL 9.1
- Symfony 1 (a PHP framework for web application)
- JQuery and several auxiliary JavaScript libraries are also widely used (Leaflet for maps, QTip for dialog and pop modals windows)
- Apache web server
- Jaspers Report for reports

Meanwhile, (2014-2016), DaRWIN2 was also being used at the RMCA to import vertebrate collections.

The possibility to rebuild entirely the interface of DaRWIN, that had been developed in Symfony 1.4 in 2013-2016, has been considered before being ruled out at the beginning of the project. The project finally stick with the current version of the framework, as re-building it in a newer library would represent a consequent financial investment, consuming time and manpower without offering additional functionalities. This would also have represented a rupture for the users at RMCA and RBINS; that would need to be trained on newer systems, while they were already familiar with the system. New functionalities would progressively be appended to the existing DaRWIN system.

2. METHODOLOGY

During the initial year of the project, DaRWIN has been compared with other solutions, like Specify and PlutoF. We also discussed the possibility of using it with living collections for AMP Meise.
A public version of Specify has been installed at RMCA and evaluated by scientific staff of the three partner institutions. The reasons to keep using DaRWIN were :

- The data model of Specify was more complex than the one of DaRWIN (130-140 tables against 90-100 for DaRWIN)
- The DaRWIN module to import external data was more powerful (allowing to pause and restart the importation of data when working with big datasets)
- The design and user experience of the interface were not better than those of DaRWIN

- The technical architecture of Specify was complex, with a core in Java connecting a server in Python, making it complex to maintain and evolve

Meise Botanical Garden also considered the possibility to use several other CMS systems:

- PlutoF (https://www.natmuseum.ut.ee/et/content/plutof), developed by the University of Tartu (Estonia), which was a cloud-based solution intermediate between a collection management system and an Internet API to publish and exchange data (such the GBIF IPT) and
- Botalista (https://botalista.community/), a collection management system developed by the Botanic Garden of Geneva (a meeting gathering Geneva and RMCA staff has been organized in 2017).

It appeared that one of the main assets of DaRWIN was its mechanism to import external data that allowed uniqueness control and merge (at the time of the meeting of this meeting, only for specimen data), that allows to work in several sessions, by stopping and resuming the importation work at hand. However, DaRWIN has not been used by Meise as its data model couldn’t easily be applied to living seed collections. APM Meise finally used BGBase.

A star diagram has also been produced, with DaRWIN and MARS standing at its center, listing other systems using or providing data, and the technical exchange standards. A Table estimating the Manpower (in Person/Months) had been associated to each tasks

The developments in DaRWIN that have been undertaken within the framework of NaturalHeritage focused on 6 aspects :

1. Reduce the complexity of the mechanism of importing and cleaning external data into DaRWIN, in order to make it more scalable and flexible

2. Improve functionalities related to data exports, like the reporting functionalities of DaRWIN, and its interaction with external tools (i.e. thermal labels), by making it more self-contained and able to cope with large datasets

3. Integrate DaRWIN within *linked data”* European networks, by creating a JSON representation of the data (similar to a REST API) and a mechanism with stable and permanent identifiers. This would also improve the possibility to cite DaRWIN data into scientific papers. A data output to map the data in the DaRWINCore archive format would also be implemented in DaRWIN

4. Simplify and rationalize the data model, by improving its normalization, to avoid data redundancy and ease the input of data by technical staff

5. Adding a more powerful geographical layer, allowing to
    a. clean geographical data and to handle non-structured text
    b. Handle semi-structured geographical data in text format, with a

multi-lingual aspect

c. Compare the geographical information in DaRWIN with external sources (OpenStreetMap) for administrative boundaries but also ecological areas

6. Implement changes requested by users of both institutions, if they can be implemented in a realistic time, without changing the architecture of Darwin, and fix the bugs they encounter. These changes are often related to the user experience of the interface.

This work could be considered as a *reverse engineering* process, as the team having initially developed the tool had left in 2017, though we were in contact with Paul-André Duchesnes until his departure, who always kindly and constructively answered to our questions

The search engine of DaRWIN has been re-engineered, by keeping the same interface). This made it less complex and more powerful, and improved its speed and ability to handle and export large batches of data. DaRWIN has then become more self-contained (less dependent from third party applications) than before, making it easier to maintain and to install. The above-mentioned best practices were also relevant to make Darwin scalable and integrate it in future system managing collection and biological data, making it able to interact efficiently with networked *Big Data* architecture. Darwin would play the roles of 1) the service keeping the primary and ecological data and 2) of a platform to integrate clean, reconcile and clean data coming from external sources. This would help to integrate DaRWIN into national and international scientific IT systems that have to deal with larger amounts of data. Engineering patterns have been elaborated while reworking DaRWIN :

1. The usage of accumulators that are very useful to speed-up complex queries. It allows the system to return large amounts of records (up to 500 000) and fields (up to 150 columns) in a reasonably good time (between a few seconds and five minutes for very large datasets). These accumulators are often preferable to the classic queries using standard foreign keys and *join* statement: they increase in realistic proportions the memory consumption of the system, but speed-up (sometimes exponentially) the speed of queries hey are more verbose than the classic queries based on JOIN statements, but more readable and self-documented. The increased consumption in memory is counterbalanced by a much more gain in time complexity. In a PostgreSQL context, accumulators are implemented as *WITH* statements[1], that can be compared to temporary and in-memory aliased views that are immediately reused by other queries (as pseudo-tables) in a single transaction and immediately deallocated. This actually simplifies the code of DaRWIN, as chaining *WITH* queries often spare the need to develop more complex stored procedures and functions that are hard to debug. *With* statements can also be natively defined in the PHP part of DaRWIN, or as SQL views. The gain in speed allowed the development of more complex and scientifically meaningful reporting mechanisms on top of DaRWIN data. This

[1] https://www.postgresql.org/docs/9.6/queries-with.html

represented an opportunity to get more familiar and use engineering practices associated to *dynamic programming*[2] in the context of large collection databases, that can transform case of exponential complexity into linear problems[3]

2. The usage of asynchronous queries, running in background, when large amounts of data have to be imported into and/or exported from DaRWIN. This was already the case in the original version of DaRWIN, when importing external data, but this asynchronous part was not directly linked to the web interface and had to be triggered by a console (often remote SSH) connection, which took the import and reporting functionalities away from non-IT users. A custom-made template for semaphore and notification, allowing to synchronize background queries with the web interface, using flags exposed on the web in JSON format and able to be detected by AJAX libraries (like JQuery) has been developed within the project. This allows to develop more powerful reporting and importing functionalities available from the web interface, enlarging the user community of DaRWIN. The same tool is now made relevant for collection technicians and their scientific managers.

3. Reliability and scalability have been increased by using simple, consistent and homogeneous data structures inside DaRWIN. Darwin originally required converting imported data into an intermediate XML file, based on the ABCD[4] schema, which is hierarchical, and had to be converted again into a tabular structure when stored in DaRWIN. The ABCD version couldn't be published on the Internet (e.g. to publish the data to GBIF). The conversion was done by a Visual Basic Macro embedded in a specific Excel worksheet. This needlessly increased the complexity of the system and made it much harder to maintain (as bugs were harder to locate) and expand (as each update of the data model required to modify several and test again separated applications written in different languages). It also introduced a high level of coupling and dependency between DaRWIN and the Excel macro. Simplifying this structure, by using data in tabular format in any part of DaRWIN (import, internal conversion and quality check and export) made both the system and workflow much more reliable and faster, and easier to update. In the same vein, it is also much more interesting (for portability and performance reasons) to handle simple and binary-readable formats like tab-delimited files or CSV (rather than Excel or XML) files when importing data into the system and generating reports. These formats are flexible, interoperable, make the system more self-contained, which is interesting for partners having limited IT infrastructure like in developing countries. As the database is itself made of tabular structures, generating them is fast and memory-efficient.

One of the most significant enhancement brought to DaRWIN is the improved

[2] https://en.wikipedia.org/wiki/Dynamic_programming and https://en.wikipedia.org/wiki/Greedy_algorithm
[3] For instance, by avoiding recursion (and exponential iteration on Cartesian product) when a table is called several times in the JOIN statements of a query, which often happen in DaRWIN
[4] https://abcd.tdwg.org/

integration in linked data networks and the usage of standards for data, making it usable and interoperable in international projects:

1. standards for the identification of physical and moral persons are natively managed (ORCID, GrSciColl, GRID, etc...)
2. the bibliographical module has been expanded and can handle DOI and other standards
3. The system also has a mechanism to allocate UUID (uniform unique identifiers) to specimens and keep them in memory. This makes DaRWIN compliant with European data architectures like CETAF. This also improve the possibility to cite DaRWIN data in scientific articles
4. data are exposed on the Internet by an HTTP GET / JSON interface, that can uses several identifiers as locator (internal technical ID of DaRWIN, main collection numbers, UUID for specimens, ORCID or other IDS for persons), making data linkable and retrievable for in knowledge databases
5. tools have been developed by Franck Theeten and Thomas Vandenberghe to export DaRWIN data in formats used by GBIF (EML scripts and embedded materialized views in SQL), a solution which could be reused for other data networks (such as WoRMS, DISSCO)
6. geographical data car also use external sources like OpenStreetMap, GeoNames, OGC WMS for quality check, calculation of uncertainty radius, linkage to polygons and area and translation
7. DaRWIN is also used to store searchable DNA tags stored in GenBank
8. An external tool has also been developed that can compare, check and reconcile DaRWIN taxonomic hierarchies with several available API (DARWIN, GBIF, Catalogue of Life, WORMS for marine species, Fishbase)[5].

3. INFRASTRUCTURE

Three test servers have been installed, one at RMCA and two at RBINS (one for the PostgreSQL DB and one for the Symfony backend). A GitHub repository has also been created on the main repository page of the RBINS, thanks to the ICT service of RBINS [6]

3.1 Technical update of DaRWIN core components

In 2019, the RBINS ICT infrastructure updated the PostgreSQL server from version 9.1 to version 9.6 In 2020, a server with PostgreSQL 12 has been successfully installed at RMCA to import wood biology data.

The change of version initially created bugs, as the syntax to handle *hstore* (key/value array) was broken. However, problematic segments of code could be readily identified and modified. The change of version was actually interesting as the newer versions of PostgreSQL are faster and more memory efficient than 9.1.(introduction of System V

[5] http://naturalheritage.africamuseum.be/natural_heritage_webservice/taxonomy/
[6] https://github.com/naturalsciences/natural_heritage_darwin/

and better handling of parallelism and multiple -CPUs architecture).
They also introduced several interesting libraries, among them :

- A native JSON data type
- An engine to write stored procedures and functions in Python language (Pl/Python [7]). Combined with the JSON data type, it allowed writing an HTTP REST client that can directly be used in PostgreSQL tables and functions, providing an interesting level of performance when comparing the data from DaRWIN with external resources published on the Internet. We used this to connect geographical web services such as GeoNames and OverPass-Turbo and OpenStreetMap APIs, to check and/or complete the geographical information in DaRWIN. This solution actually enables DaRWIN to associate coordinates to geographic information that were initially not geo-referenced, and to store geographical areas (counties, provinces, but also lakes and desert etc...) and lines (roads and rivers) available from OpenStreetMap, and not only point data. The embedded Python client could also connect other services (e.g. GBIF API and Catalog of Life API for taxonomy), Sketchfab API for 3D models.
- Materialized views, that can be very useful to map DaRWIN data to exchange data standard, such as DaRWINCore Archive or EML are now available
- The *Foreign Data Wrapper* are functional. This is a low level broker that allows to map remote external resources (Spreadsheet documents, other databases' engine such as Oracle, MySQL and MS Server, other PostgreSQL servers) to pseudo-tables present in a PostgreSQL schema. This solution represents a very valuable alternative to JDBC clients. It is much faster than JDBC (as it runs at a lower technical level), and the external data source can very easily be mapped to SQL views and joined with other tables. It spares the need to develop an external script, as, once the connection is done, any exchange of data between servers can be done directly via “classic” SQL statements. This is also an interesting way to link data present in external, possibly old legacy, servers to the libraries present in PostgreSQL (like PostGIS for geographical treatments and linguistic libraries). We extensively used *foreign data wrappers* to migrate the data from the original PostgreSQL data model at the RBINS to the reworked one, that has another normalization of collecting localities, in August 2019. The whole migration process, prepared upstream with developments, took about 2 hours to complete, and was reproducible as its logic (migration and integrity checks) was embedded in views and stored procedures stored in a specific database schema.

One of the main limitations of the original version of Symfony was that it was only compliant with PHP 5, which is much slower than PHP 7 that appeared in 2016-2017. In July 2019, we identified a port of the Symfony 1.4 framework in PHP 7.2 developed by the IT developed of the French weekly *L’Express,* available on GitHub[8]*.* This forked version of Symfony was successfully adopted in August 2019, while performing a migration of DaRWIN at the RBINS. The name of a few PHP class constructors, especially those calling the Doctrine ORM (SQL query wrapper) had to be modified, but this could be done systematically and in a semi-automatic way by using the Bash *grep*

[7] https://www.postgresql.org/docs/9.0/plpython-funcs.html
[8] https://packagist.org/packages/friendsofsymfony1/symfony1

and *sed* commands on the whole code folder
All in all, the usage of the *LExpress* fork greatly enhanced the responsiveness of DaRWIN for users. It also improves the scalability, the security and the life-expectancy of DaRWIN. No downtime related to this framework occurred since it has been installed. However, some PHP 7 Linux libraries have to be installed with custom wrappers to follow the PHP 5 naming convention [9]
We also replaced, when possible, the external Jaspers reporting tool by embedded tasks directly outputting reports in CSV (tab-delimited) formats. This makes DaRWIN more self-contained as it is not dependent anymore from a third-party tool running on a specific Java server. The reports are also made more detailed and descriptive as they are directly dependent from SQL views or queries, that are the only elements to work on if the report structure needs to be modified and expanded. Besides, they can handle a virtually unlimited amount of data as they are implemented as asynchronous background tasks, associated with pagination that help to limit the CPU and RAM consumption. DaRWIN now uses semaphore and locks (sent in JSON format to an Ajax script placed in a specific page) to notify the user when a resource is ready and enable download, and avoid concurrent execution of the same process. This feature is very useful to generate yearly statistical reports of the activity in the database or the specificities of a collection (amount of types, taxa). It also ensures that virtually all data inserted in the system can be exported back in a spreadsheet format and reused in other systems, at user's request.

4. RESULTS AND RECOMMENDATIONS

4.1 Renormalization of localities and collecting dates

The data model of DaRWIN was generally sound, but the way localities and collecting dates were normalized led encoders to often duplicate geographical data. Collecting stations and collecting dates were merged in a single table, called *GTU (Geo-Temporal Units)*. When several consecutive collecting events occurred at the same position, or concerned the same non-georeferenced locality that hay may have been described at a very generic level, the record had to be duplicated with only date values changing. This has been perceived as a tedious operation by encoders, who had to browse to several records in lists when associating newly created specimens or observations to their geographical origin. Even the station number didn't guarantee the uniqueness of data.

[9] Especially the APC PHP cache, renamed APCU

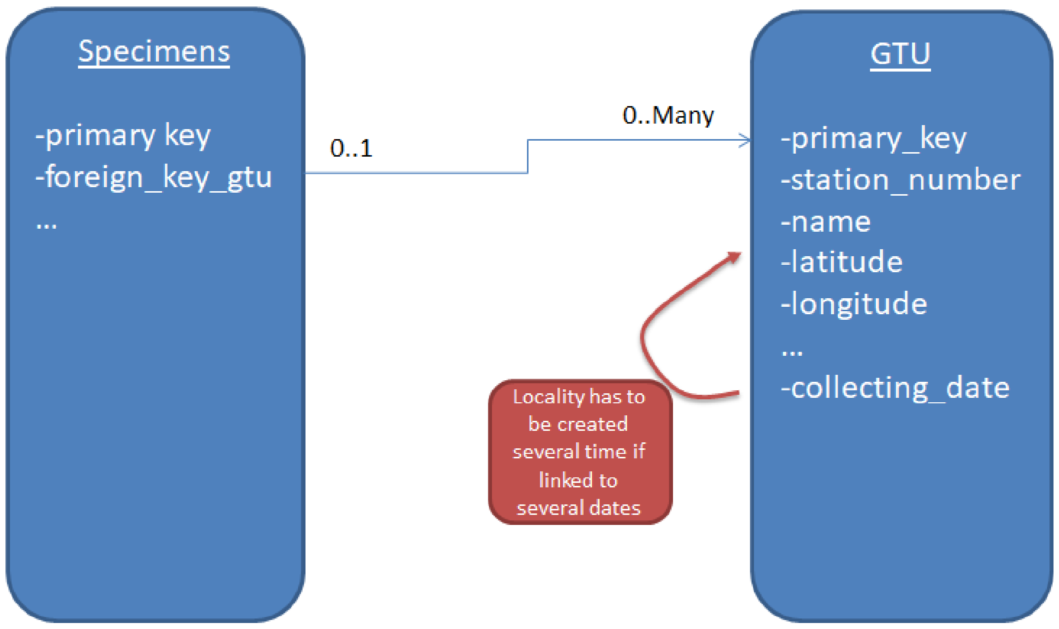


Figure 1. Initial normalization of dates

To fix this issue, we first considered that the collecting dates were an attribute of the specimen rather than the locality, and moved them to the "specimen" table. This improved the normalization of most of the values already present in DaRWIN. However, we were at the same time working on the import into DaRWIN of data from the old MISTA database (also developed by the team having developed DaRWIN), storing data on Belgian marine missions in Antarctica. The collecting place could be a mobile platform, like trawling boats using nets sprawling on the Ocean floor, that are punctually hauled on the boat. In this situation, the locality has indeed temporal characteristics. Actually the original normalization fits very well with data where the expedition information is central and can be related to a boat, while the second one is adapted to more static collection data, associated to a geographical point, which still forms the majority of data in DaRWIN.

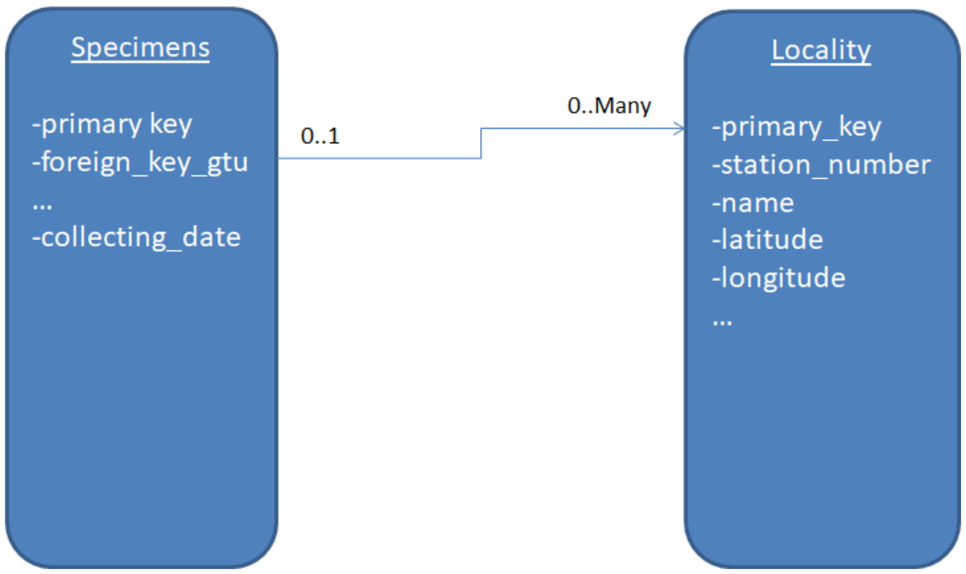


Figure 2. First proposal for renormalization

We finally adopted an intermediate solution that can work in both situations, by considering that the collecting data is neither a characteristic of the sampling location, nor the specimen, but of the many-to-many relationship between them (see fig.3). A third table called *TemporalInformation* has been introduced. It has two foreign keys with a zero-to-one relationship to the specimen (a specimen can have an unknown or one known collecting date[10]) and a one-to-many relationship to the locality (a locality can correspond to zero, one or several collecting events occurring at different times, each of them related or not to a specimen). It is now possible to associate several dates to a locality even when no specimens have been encoded yet, and therefore to register detailed expedition data that are not yet linked to specimens.

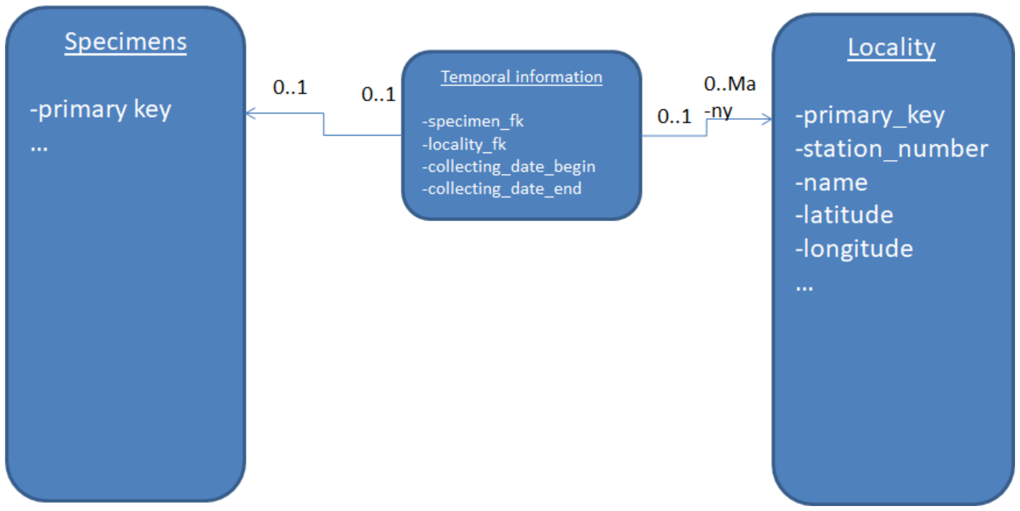
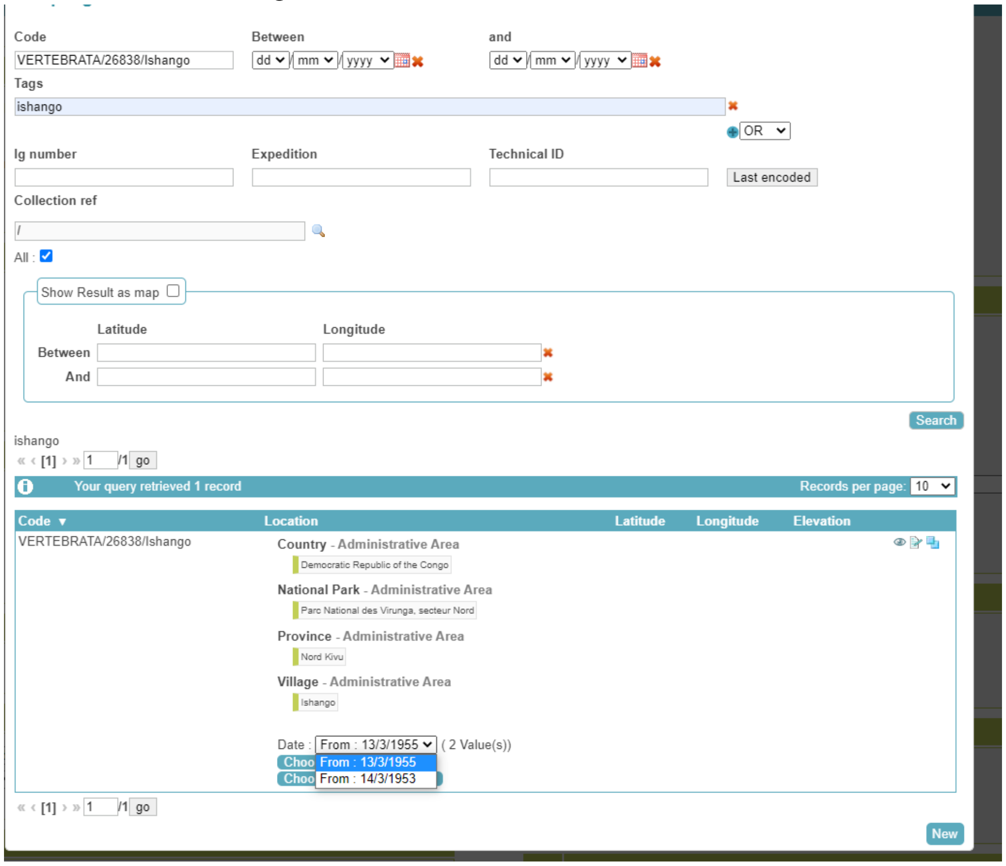


Figure 3. Adopted solution

[10] There are actually collecting period, with a beginning and end data and a notion of daily, monthly or yearly accuracy

The geographical data have been re-normalized (i.e. duplicates have been removed) within this new model, and an additional drop-down field allowing users to pick a date up amongst several possibilities has been appended to the interface linking newly created specimens to localities. Collecting date can also be created directly at the level of the specimen. This actually helps to limit the amount of geographical data stored in DaRWIN and makes its usage more intuitive.



Code: VERTEBRATA/26838/Ishango
Between: dd / mm / yyyy
and: dd / mm / yyyy
Tags: ishango
OR
Ig number
Expedition
Technical ID
Last encoded
Collection ref: /
All : ✓
Show Result as map

	Latitude	Longitude
Between		
And		

Search

ishango
« ‹ [1] › » 1 /1 go
Your query retrieved 1 record — Records per page: 10

Code ▼	Location	Latitude	Longitude	Elevation
VERTEBRATA/26838/Ishango	Country - Administrative Area: Democratic Republic of the Congo; National Park - Administrative Area: Parc National des Virunga, secteur Nord; Province - Administrative Area: Nord Kivu; Village - Administrative Area: Ishango; Date : From : 13/3/1955 (2 Value(s)) — From : 13/3/1955; From : 14/3/1953			

« ‹ [1] › » 1 /1 go
New

Figure 4. Interface for locality and date selection

4.2 Enhanced geographical information

The GIS part of DaRWIN has also been considerably reworked and expanded, taking advantage of the possibilities offered by PostGIS (the geographical module of PostgreSQL).

4.2.1 Integration of geographical OGC Web Map server

Maps in the web Interface of DaRWIN, initially using the Leaflet JavaScript library, have been replaced by OpenLayers[11], a well-maintained and powerful Open-Source library, which is compliant with several geographical formats (OGG WMS, OGC WFS, Well-Known Text) and map providers (Google Map, Bing, OpenStreetMap, tiled layers, IIIF images, OGC WMS servers). It also supports multiple projections, full screen views

[11] https://openlayers.org/

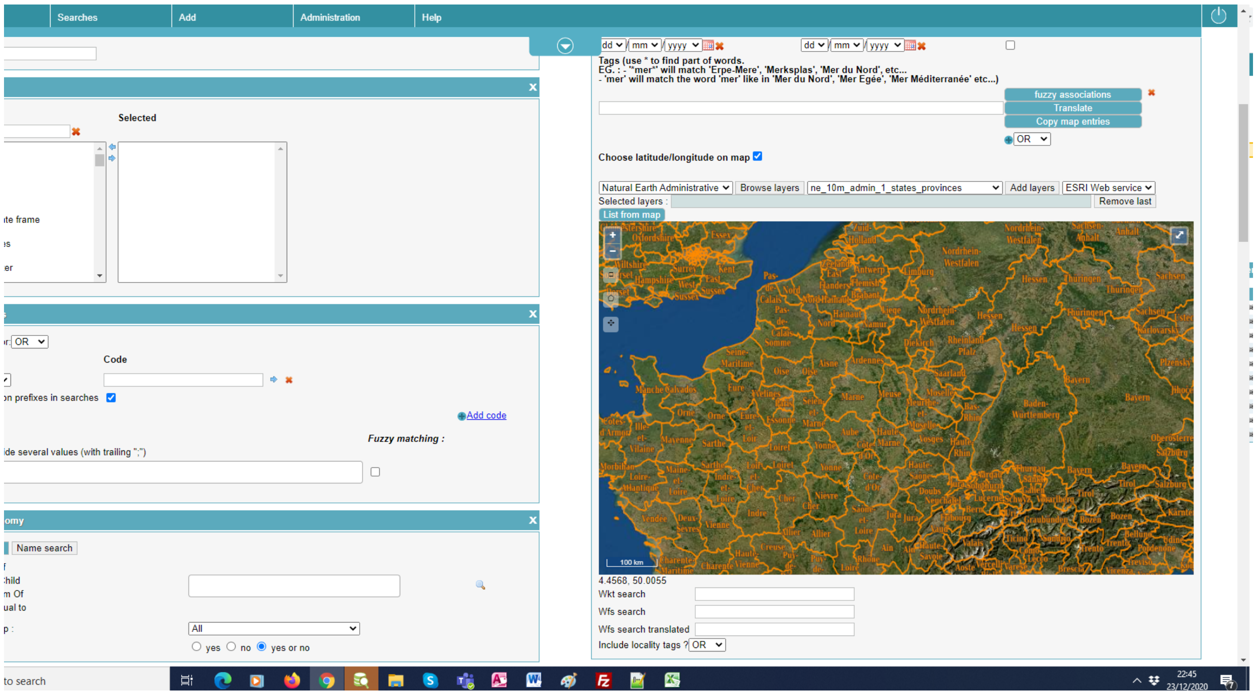
and can collaborate with JQuery. Its documentation and API are complete and well-documented.
One of the new features in DaRWIN is the possibility to search geo-referenced data that are not only within a squared bounding-box, and also in irregular polygons (that can be an ecological area, a municipality, a lake, a coastal area). These polygons can be defined in two ways:

1. Directly drawn by the user on a map, on top of a satellite or OpenStreetMap background
2. Selected from a reference and accurate layers (displaying administrative boundaries, oceans and sea, natural area such as mountain range or desserts…) that are displayed as semi-transparent foreground on the maps, by a single click

A specific database schema has been created to store these layers. Two valuable and free sources of layers, in Shapefile or GeoJSON formats, have been used :

1. *NaturalEarth*[12] a static repository of global maps, available in ESRI Shapefile format, that is supported by *NACIS (North American Cartographic Information Society)* and used by media such as *Springer* and *The Washington Post*. This source is interesting as it contains polygons of administrative areas (like the countries and national regional subdivisions) but also natural areas such as deserts, oceans and seas, mountain ranges, basins. These layers have been imported in a PostGIS table, and sometimes, split into different categories by using SQL views
2. We also used data from the Open-Source and citizen-science project *OpenStreetMap,* downloaded via *OverpassTurbo* API[13], to get the administrative subdivisions and municipalities of Belgium, and the river of Belgium. Chunks of river have been merged by using a custom-made SQL query that used the name of the river to reunite the segment in continuous or semi-continuous lines.

[12] https://www.naturalearthdata.com/
[13] https://overpass-turbo.eu/
,https://wiki.openstreetmap.org/wiki/Overpass_turbo/Extended_Overpass_Turbo_Queries



Tags (use * to find part of words.
EG. : - '*mer*' will match 'Erpe-Mere', 'Merksplas', 'Mer du Nord', etc...
- 'mer' will match the word 'mer' like in 'Mer du Nord', 'Mer Egée', 'Mer Méditerranée' etc...)

Choose latitude/longitude on map ✓

Natural Earth Administrative | Browse layers | ne_10m_admin_1_states_provinces | Add layers | ESRI Web service

Selected layers :

ɪr: OR

ɔn prefixes in searches ✓

Fuzzy matching :

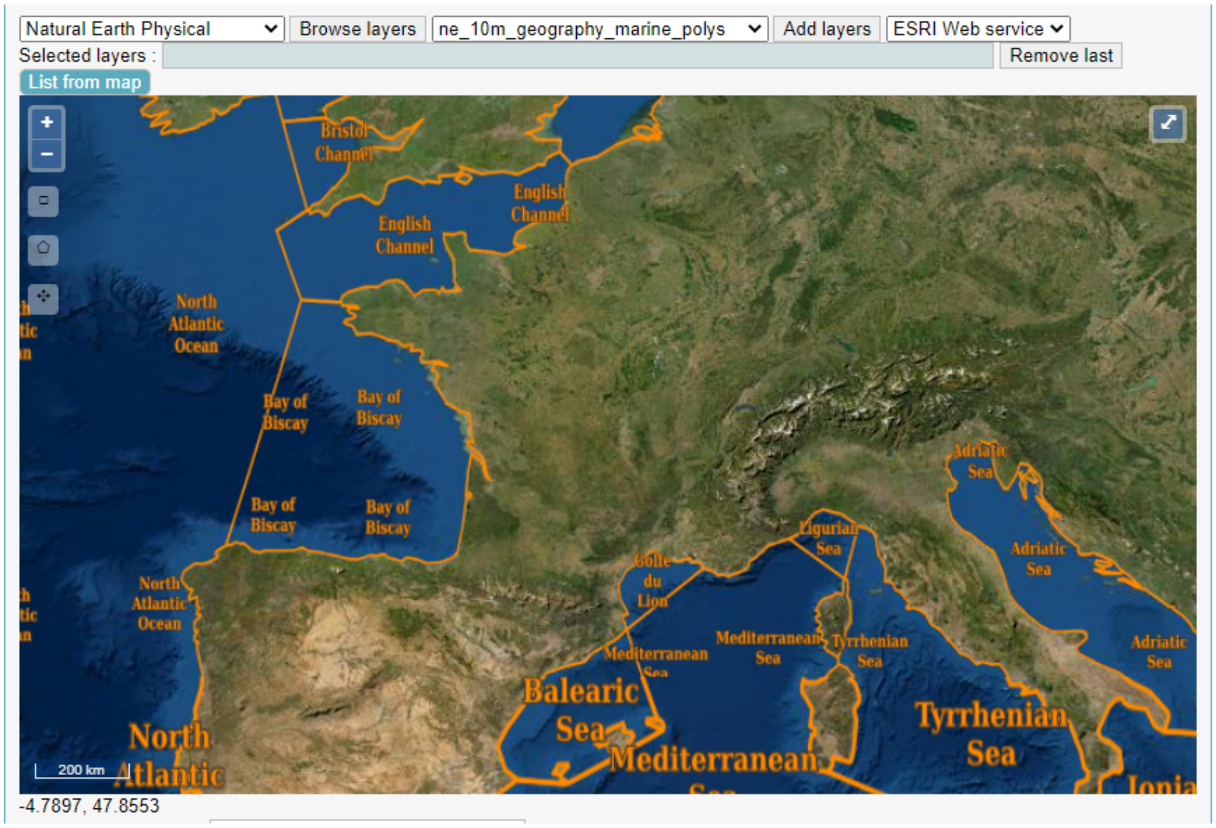
ɪde several values (with trailing ";")

All

yes no [✓ yes or no]

4.4568, 50.0055
Wkt search
Wfs search
Wfs search translated
Include locality tags ? OR

Figure 5. Regional subdivision of Belgium and Western-Europe
From the NaturalEarth layer In DaRWIN query interface



Natural Earth Physical | Browse layers | ne_10m_geography_marine_polys | Add layers | ESRI Web service

Selected layers :

-4.7897, 47.8553

Figure 6. Marine areas from Western Europe from Natural Earth
(background satellite layer: ESRI tile server)

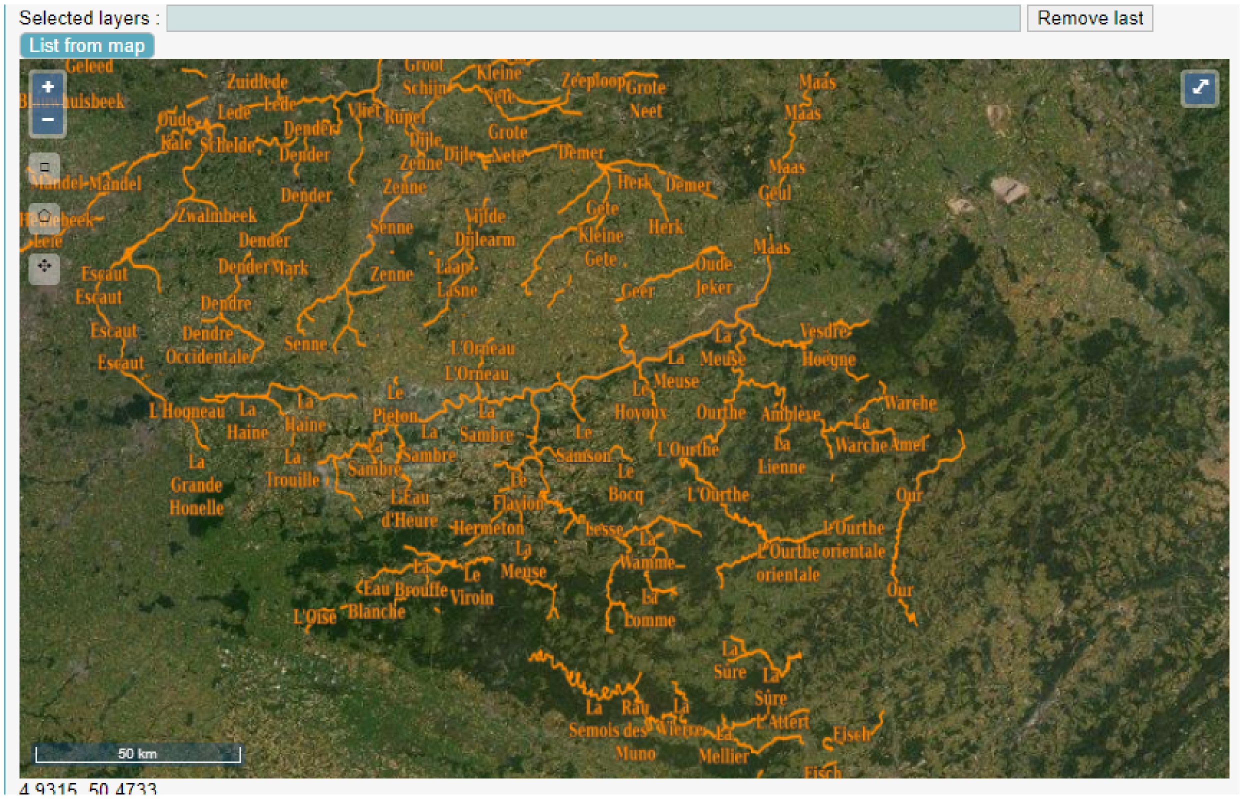


Figure 7. Belgian Rivers from Open-Street Map

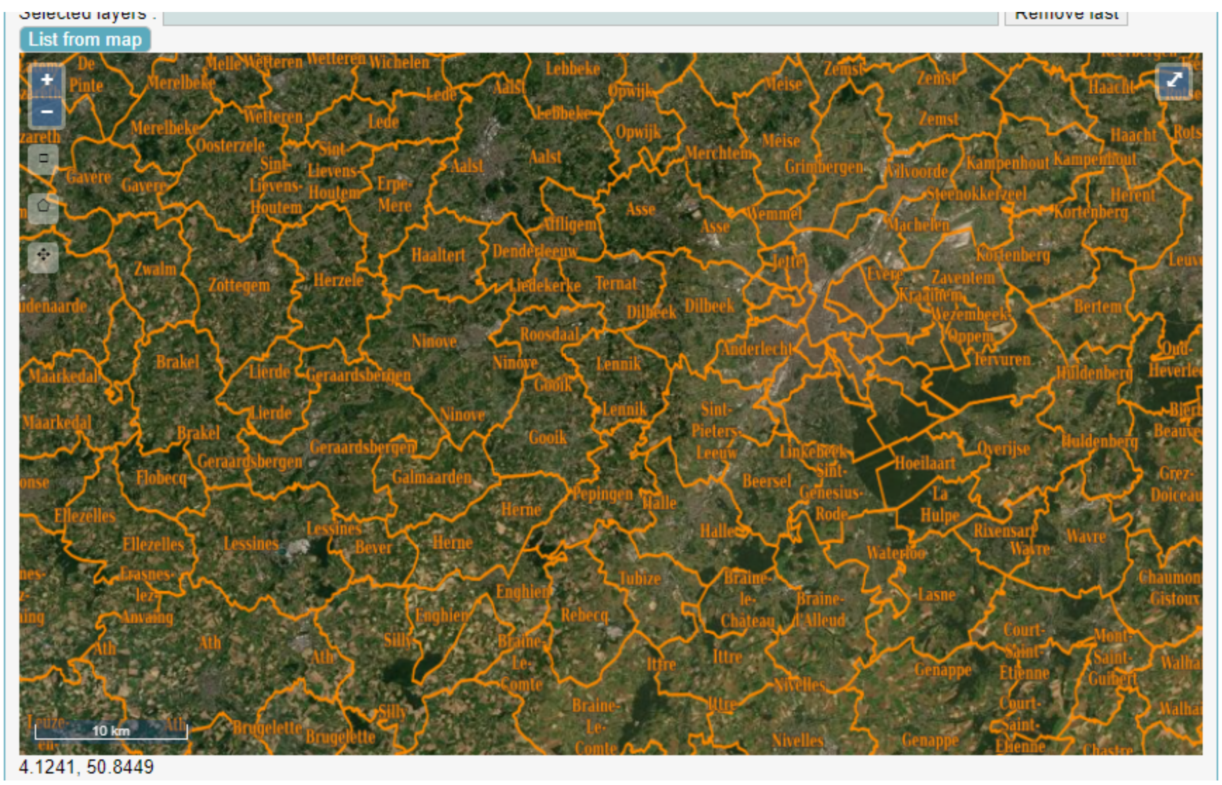
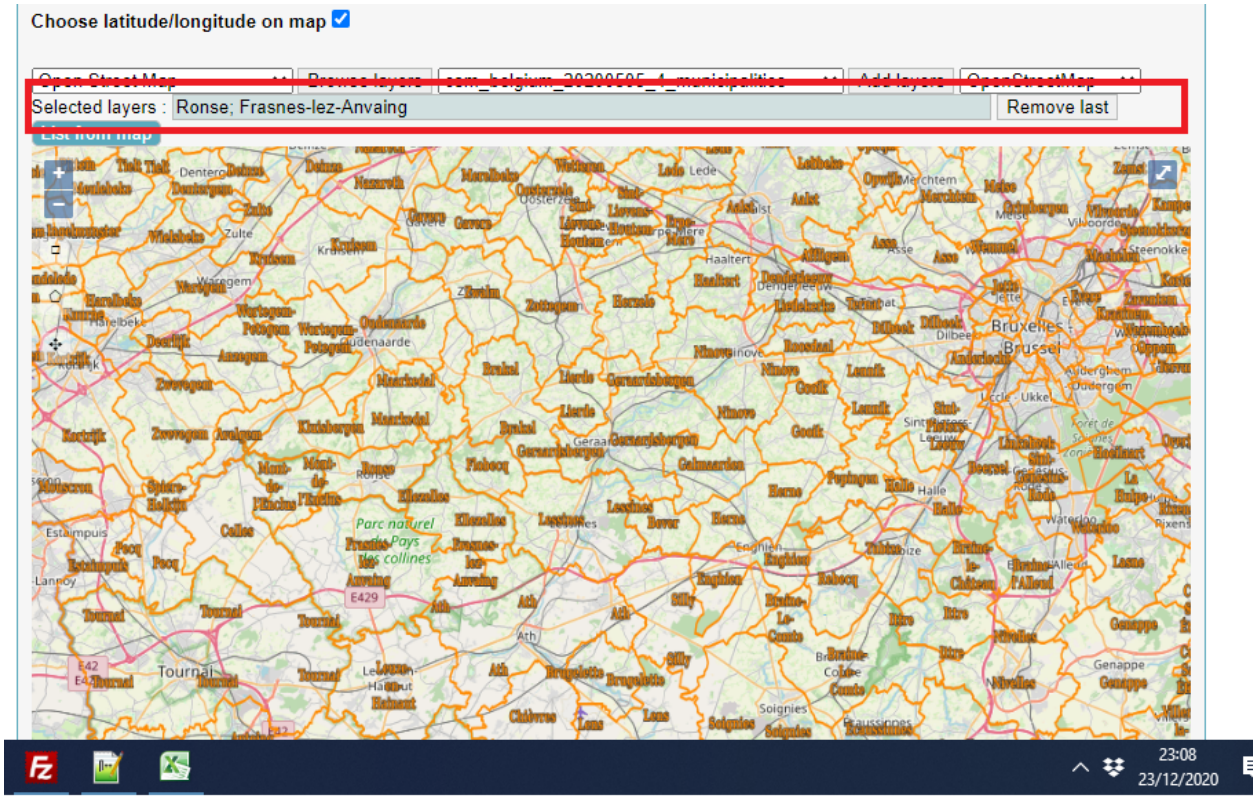


Figure 8. Municipal (communal) boundaries of Belgium from OpenStreetMap

These layers are stored in PostGIS, and made available in the query interface of DaRWIN by using an Open-Source GeoServer[14] WMS (*Web Map Server*[15]), which is "*proxyfied"* by the DaRWIN server of the RBINS. The user can select the layer by using two drop-down lists, the first one with the source of the layer and the second one with the list of available layers. A select button enables layers in the map. Users can select one or several areas simply by clicking on the selected layer, which is highlighted in orange on the map. A list contains the selected zone, with the possibility to remove the last one.


Choose latitude/longitude on map ☑

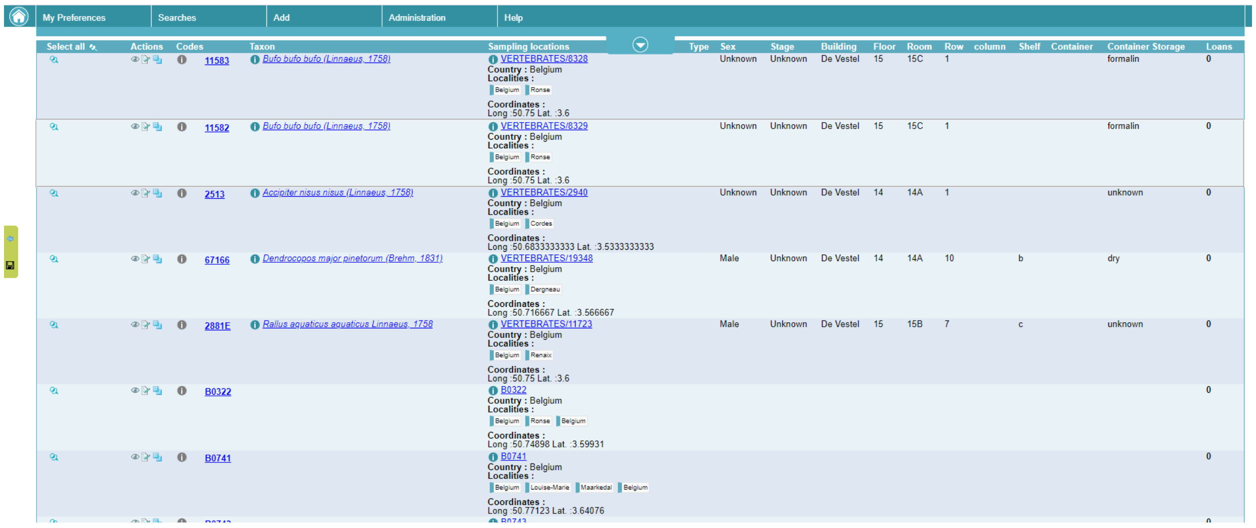
Selected layers : Ronse; Frasnes-lez-Anvaing

Figure 9. List of selected areas (Ronse and Frasnes-lez-Anvaing)

Search queries will retrieve the geo-referenced records that fall within the selected areas. User(s) can therefore select large and meaningful geographical areas just in a few clicks.

[14] https://darwin.naturalsciences.be/geoserver/web/ *GeoServer* is a reference open-source implementation of the OGC WMS and WFS protocols, dedicated to Java Server, which is relatively easy to install and manage.

[15] WMS (*Web Map Service*) is an open geographic protocol defined by the OGC (*Open Geospatial Consortium)* which standardizes the publishing of vector and raster maps on the web. This is a server side specification, which encompasses the rendering of vector maps, their projection, but also the querying of attributes associated to the map. It can interact with client libraries, often in JavaScript (like *OpenLayers*) to merge layers from several sources and create interactive and searchable maps on graphical web-clients. WMS has a sister-protocol called *WFS (Web Feature Service*) which serves the maps in full-text XML (GML), GeoJSON or KML (Google) formats.



Select all	Actions	Codes	Taxon	Sampling locations	Type	Sex	Stage	Building	Floor	Room	Row	column	Shelf	Container	Container Storage	Loans
		11583	*Bufo bufo bufo (Linnaeus, 1758)*	VERTEBRATES/8328 Country : Belgium Localities : Belgium Ronse Coordinates : Long :50.75 Lat. :3.6		Unknown	Unknown	De Vestel	15	15C	1				formalin	0
		11582	*Bufo bufo bufo (Linnaeus, 1758)*	VERTEBRATES/8329 Country : Belgium Localities : Belgium Ronse Coordinates : Long :50.75 Lat. :3.6		Unknown	Unknown	De Vestel	15	15C	1				formalin	0
		2513	*Accipiter nisus nisus (Linnaeus, 1758)*	VERTEBRATES/2940 Country : Belgium Localities : Belgium Cordes Coordinates : Long :50.6833333333 Lat. :3.5333333333		Unknown	Unknown	De Vestel	14	14A	1				unknown	0
		67166	*Dendrocopos major pinetorum (Brehm, 1831)*	VERTEBRATES/19348 Country : Belgium Localities : Belgium Dergneau Coordinates : Long :50.716667 Lat. :3.566667		Male	Unknown	De Vestel	14	14A	10		b		dry	0
		2881E	*Rallus aquaticus aquaticus Linnaeus, 1758*	VERTEBRATES/11723 Country : Belgium Localities : Belgium Renaix Coordinates : Long :50.75 Lat. :3.6		Male	Unknown	De Vestel	15	15B	7		c		unknown	0
		B0322		B0322 Country : Belgium Localities : Belgium Ronse Belgium Coordinates : Long :50.74898 Lat. :3.59931												0
		B0741		B0741 Country : Belgium Localities : Belgium Louise-Marie Maarkedal Belgium Coordinates : Long :50.77123 Lat. :3.64076												0

Figure 10. List of specimen (with coordinates) collected in Ronse and Frasnes-lez-Anvaing in DaRWIN (see Figure 17 for the corresponding query)

4.2.2 Integration of free-hands polygons and lines in the query interface

Alternatively, users can draw their own polygons by hand in the query interface, by using buttons in the top-left corner of the maps. Left-clicking the mouse defines the path, while right-clicking allows to remove the last point without restarting from scratch. Double left click closes the shape

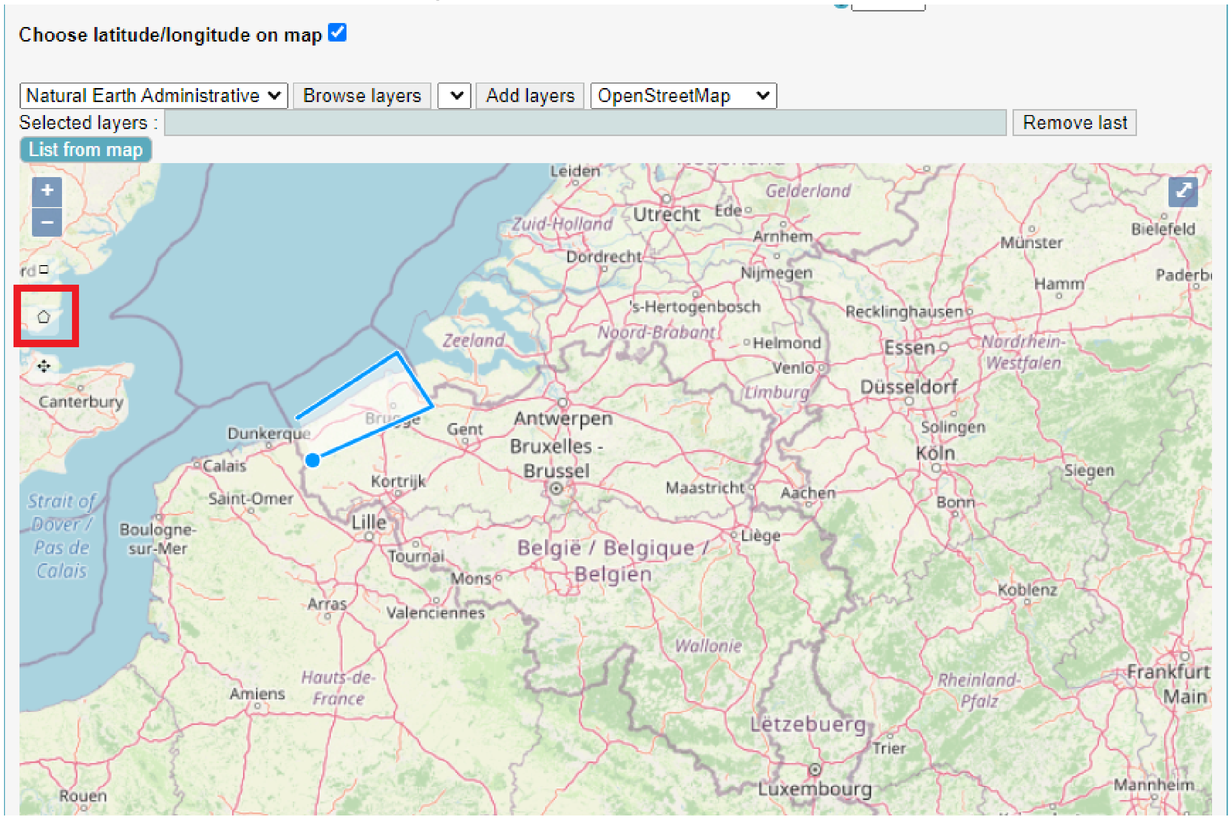


Figure 11. Manual highlight of the Belgian Coastal region

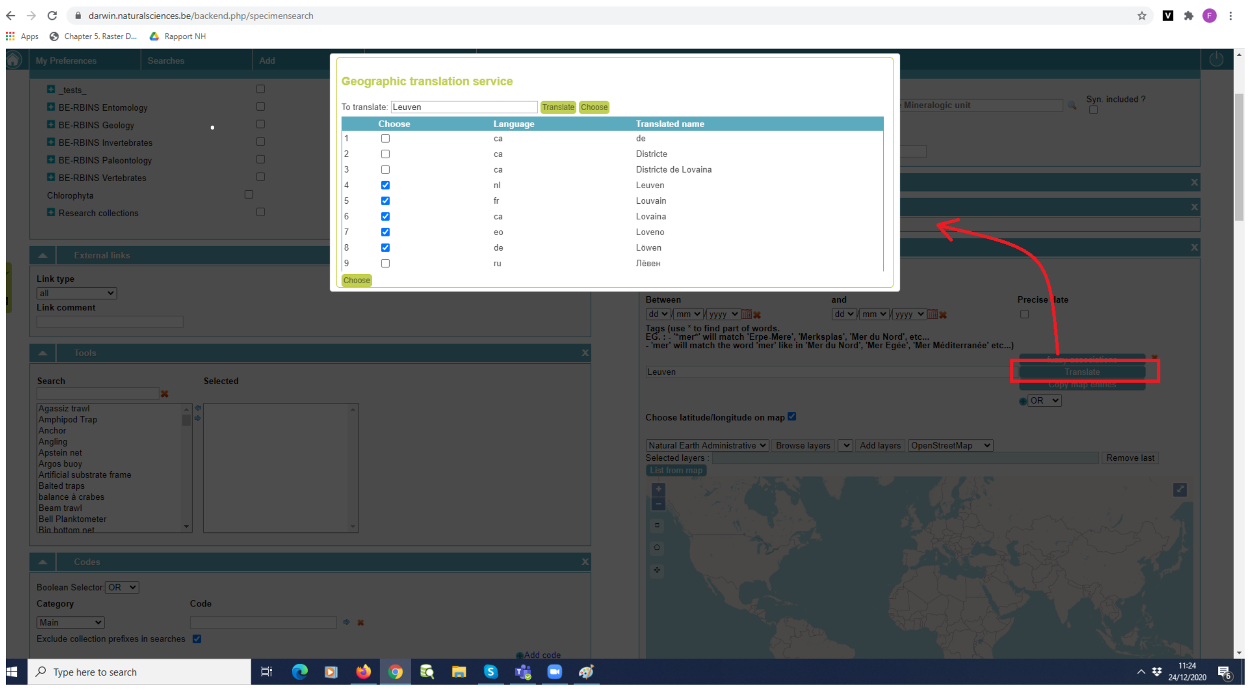
Technically, both the OGC queries and free-hand queries send the coordinates of polygons to the PostGIS database in Well-Known Text format, which is then parsed by the Symfony backend to build the SQL query. A PL/PgSQL function is defined to map the polygons of the OGC layers in the main schema of DaRWIN. This means that the OGC layers exposed to the GeoServer WMS need to be stored in the same database as DaRWIN (a remote connection using the OGC WMS only would not work).

4.2.3 Satellite background layers

As displayed above, the OpenLayers client of DaRWIN can display both OpenStreetMap layers, and satellite background layers. A drop-down list allows changing the background layer. Initially, the satellite background layers used a free version of Microsoft BING, but it has been replaced by the ESRI XYZ (i.e pyramid-tile) server[16].

4.2.4 Translation of geographical names

The geographical module of DaRWIN features a multilingual thesaurus, generated from the translated columns of the OpenStreetMap layers downloaded from Overpass Turbo. These columns have been re-organized in i.e.a SQL view and integrated into the search interface of DaRWIN. A modal window aside from the name appears when the user requests translation of a name. It displays a list of the possible matching translations, along with the ISO 639 code of the language, with checkboxes to select the terms to be used as search criteria.



Geographic translation service

To translate: Leuven

	Choose	Language	Translated name
1	☐	ca	de
2	☐	ca	Districte
3	☐	ca	Districte de Lovaina
4	✓	nl	Leuven
5	✓	fr	Louvain
6	✓	ca	Lovaina
7	✓	eo	Loveno
8	✓	de	Löwen
9	☐	ru	Лёвен

Figure 12. Translation of "Leuven" in DaRWIN

[16] Access point is https://services.arcgisonline.com/ArcGIS/rest/services

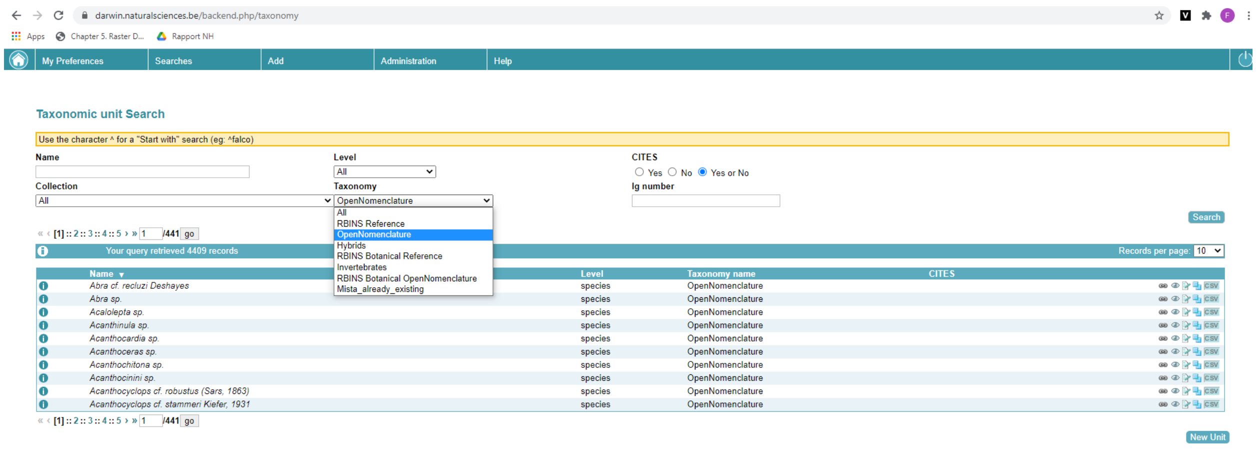
This functionality allows users to rapidly translate geographical search criteria, while still having a certain control and degree of freedom to judge the relevancy of terms. The concept is to link and complete the search keywords of the research, rather than standardizing (and correcting) the original data, which is time-consuming, and could introduce errors in the original data.
The same technical architecture could be used to control and geo-reference manually old data by giving geographical coordinates (point, line or polygons) to text description, as the OpenStreetMap layers feature a geographical field. This interface could be transformed from a search interface into a validation and georeferencing interface. A test interface has been developed during Summer 2020, but this feature has not been completed and implemented.

4.2.5 Automatic conversion DMS/ DD coordinates

Both the input interface of DaRWIN and the importation template for localities feature mechanisms to transform DMS (Degree Minutes Seconds) coordinates into Decimal Degrees coordinates (which is the most usual coordinate format in GIS), while still keeping the DMS description in an auxiliary text field. The DMS values can be placed in a single text field in the template (no need to split the degrees, minutes and seconds into separate fields) as long as they follow a consistent writing patterns that uses the conventional delimiters (° for degrees, ´ for minutes and ´´ for seconds) and feature the N, S, W, E signs placed in first or last position. This field is parsed by a regular expression available as a PL/PgSQL function.
DaRWIN assumes that coordinates follow the WGS 84 (“*EPSG:4326*”) projection. The web interface can also convert UTM coordinates using the WGS 84 ellipsoid, though.

4.3 Parallel taxonomies

Taxonomic hierarchies have been reworked so that a specific SQL table, called “*taxonomic_metadata”* has been created. These are clusters allowing to group together taxonomic hierarchies.
Each taxonomic hierarchy has a boolean “*is_reference”* field. When set to true, this field means that the hierarchy is verified and scientifically accurate. Working or incomplete hierarchies should be tagged with the reference flag set to false.
A name and its author can be only present once in each hierarchy, this mechanism represents a way to allow and manage taxonomic duplicates. The import template for taxonomies has to be linked to one specific cluster. The import template for specimens must also be checked against one specific cluster (it is possible to change the current cluster iteratively, as the check procedure operates by cycles that terminates when all specimens have been imported).
The search interface and modal windows feature a drop-down list allowing to restrict searches to specific taxonomies



Taxonomic unit Search

Use the character ^ for a "Start with" search (eg: ^falco)

Name:
Level: All
CITES: Yes No [✓ Yes or No]
Collection: All
Taxonomy: OpenNomenclature
Ig number:

Taxonomy options: All, RBINS Reference, [✓ OpenNomenclature], Hybrids, RBINS Botanical Reference, Invertebrates, RBINS Botanical OpenNomenclature, Mista_already_existing

Search

« ‹ [1] :: 2 :: 3 :: 4 :: 5 › » 1 /441 go

Your query retrieved 4409 records

Records per page: 10

Name	Level	Taxonomy name	CITES
Abra cf. recluzi Deshayes	species	OpenNomenclature	
Abra sp.	species	OpenNomenclature	
Acalolepta sp.	species	OpenNomenclature	
Acanthinula sp.	species	OpenNomenclature	
Acanthocardia sp.	species	OpenNomenclature	
Acanthoceras sp.	species	OpenNomenclature	
Acanthochitona sp.	species	OpenNomenclature	
Acanthocinini sp.	species	OpenNomenclature	
Acanthocyclops cf. robustus (Sars, 1863)	species	OpenNomenclature	
Acanthocyclops cf. stammeri Kiefer, 1931	species	OpenNomenclature	

« ‹ [1] :: 2 :: 3 :: 4 :: 5 › » 1 /441 go

New Unit

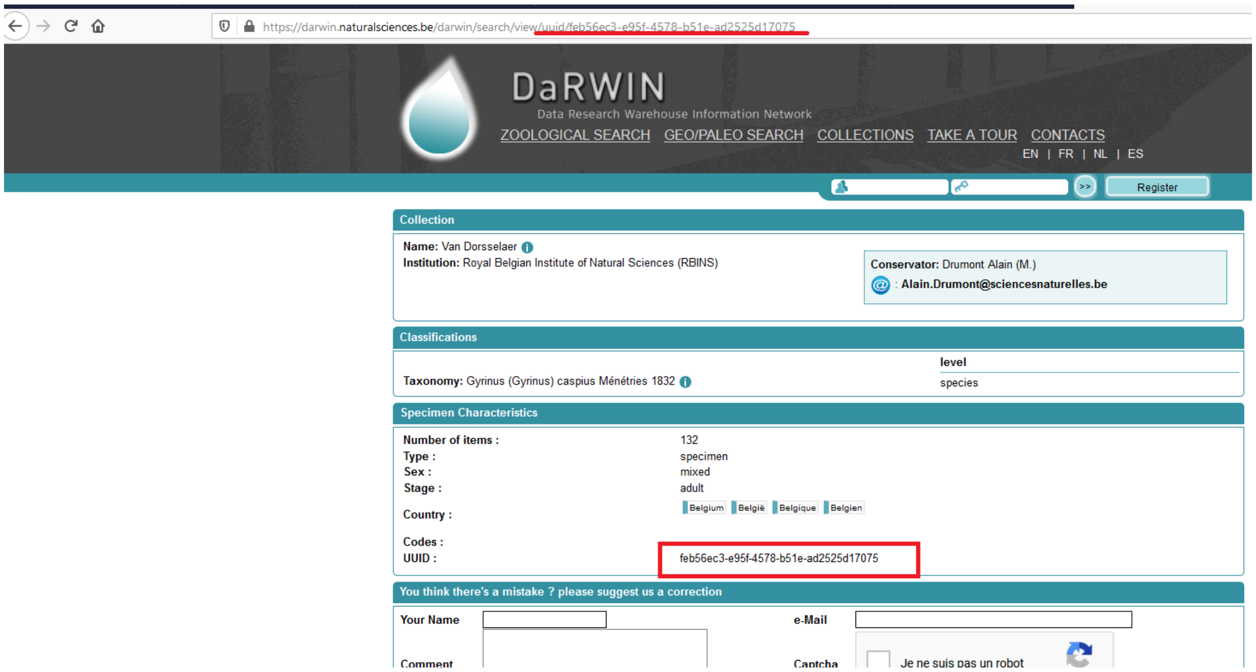
Figure 13. Taxonomic search with list of taxonomies

Users and scientists could potentially create their own taxonomic cluster and compare it with the other available classifications. A taxonomic cluster could for instance be defined for “open nomenclature” (names with annotation on their accuracy) and *nomen nodum* (provisional names used while studying potentially new taxa).
It is possible to move a name from one taxonomic cluster to another. The descending taxonomic tree moves them with the parent taxon. This system allows both a certain degree of flexibility when defining and registering taxa, and a validation mechanism that checks for duplicates.

4.4 Integration of stable identifiers

DaRWIN now features a stable identifier mechanism enabling it to be integrated in *Linked-Data* networks. This also increases the possibility to cite records from DaRWIN in scientific papers as there is a trend in European projects to use permanent identifiers, UUID being one of the possible implementations among other protocols (DOIs, handles). A specific table (*specimen_stable_ids*) has been appended in the DaRWIN data models. An entry is created in this table for any newly created records. A scheduled (*cron*) job running each evening associates a UUID (which is a native type of data in PostgreSQL) to records that are still empty. Triggers prevent deleting or modifying this value.
The PHP backend offers an access point in HTTP GET format using this UUID, which is also visible in the interface.
For RBINS the access points are :

- https://darwin.naturalsciences.be/darwin/search/view/uuid/[UUID] for the public part: e.g. https://darwin.naturalsciences.be/darwin/search/view/uuid/0acad5a0-9e9a-45b2-8e7f-dd375399f3cf
- https://darwin.naturalsciences.be/backend.php/specimen/view/uuid/[UUID] for the password-protected “backend” part: e.g. https://darwin.naturalsciences.be/backend.php/specimen/view/uuid/0acad5a0-9e9a-45b2-8e7f-dd375399f3cf


https://darwin.naturalsciences.be/darwin/search/view/uuid/feb56ec3-e95f-4578-b51e-ad2525d17075

DaRWIN
Data Research Warehouse Information Network
ZOOLOGICAL SEARCH GEO/PALEO SEARCH COLLECTIONS TAKE A TOUR CONTACTS
EN | FR | NL | ES
Register

Collection

Name: Van Dorsselaer
Institution: Royal Belgian Institute of Natural Sciences (RBINS)
Conservator: Drumont Alain (M.)
: **Alain.Drumont@sciencesnaturelles.be**

Classifications

Taxonomy: Gyrinus (Gyrinus) caspius Ménétries 1832 — **level** species

Specimen Characteristics

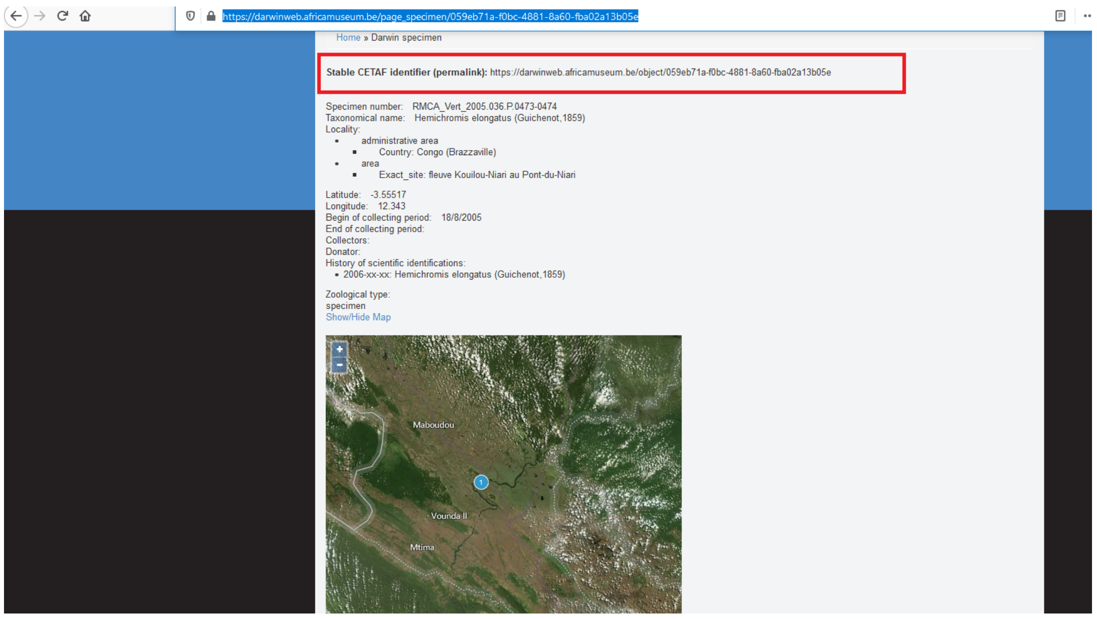
Number of items :	132
Type :	specimen
Sex :	mixed
Stage :	adult
Country :	Belgium België Belgique Belgien
Codes :	
UUID :	feb56ec3-e95f-4578-b51e-ad2525d17075

You think there's a mistake ? please suggest us a correction

Your Name **e-Mail**
Comment **Captcha** Je ne suis pas un robot

Figure 14. Public UUID view at RBINS (example)

For RMCA, the public access has the following syntax: https://darwinweb.africamuseum.be/page_specimen/[UUID]: e.g: https://darwinweb.africamuseum.be/page_specimen/059eb71a-f0bc-4881-8a60-fba02a13b05e
This links to a public web interface developed in Angular/JS and Drupal 7 as DaRWIN is here restricted to Intranet users.


https://darwinweb.africamuseum.be/page_specimen/059eb71a-f0bc-4881-8a60-fba02a13b05e

Home » Darwin specimen

Stable CETAF identifier (permalink): https://darwinweb.africamuseum.be/object/059eb71a-f0bc-4881-8a60-fba02a13b05e

Specimen number: RMCA_Vert_2005.036.P.0473-0474
Taxonomical name: Hemichromis elongatus (Guichenot,1859)
Locality:
- administrative area
  - Country: Congo (Brazzaville)
- area
  - Exact_site: fleuve Kouilou-Niari au Pont-du-Niari

Latitude: -3.55517
Longitude: 12.343
Begin of collecting period: 18/8/2005
End of collecting period:
Collectors:
Donator:
History of scientific identifications:
- 2006-xx-xx: Hemichromis elongatus (Guichenot,1859)

Zoological type:
specimen
Show/Hide Map

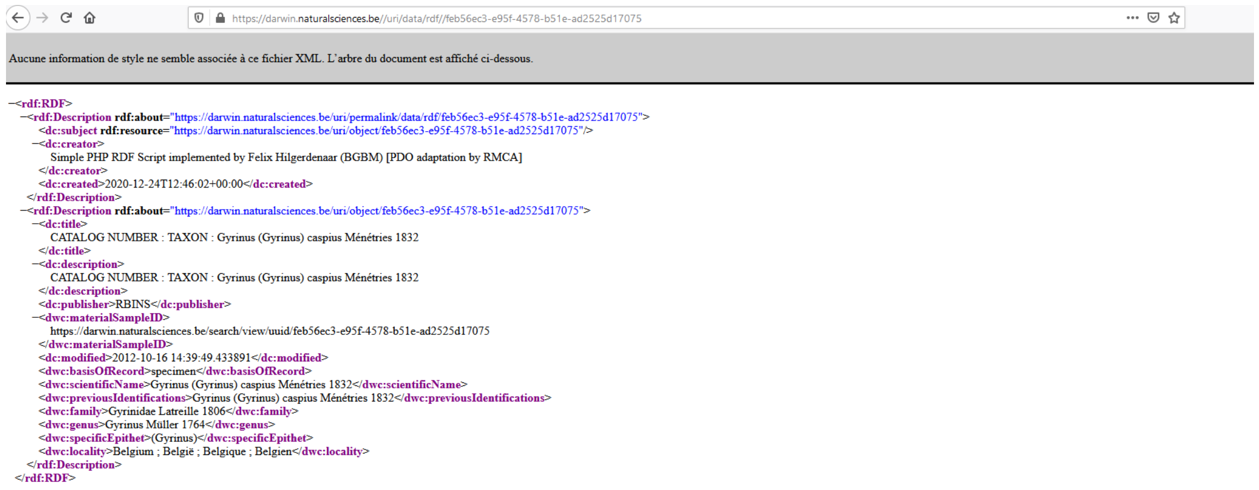
Figure 15. Public UUID view at RMCA(example)

The CETAF script for stable URL has also been adapted to both DaRWIN (RMCA and RBINS). This script used stable URLs (not limited to UUIDs , but UUID can be one of them) providing an access point to a revolver which either :

- Forwards the page to an HTML representation of the specimen
- Forwards the page to an RDF/XML page with the metadata of the specimen by content negotiations (the client mentions in the query HTTP header whether it requests the human-readable HTML page or expects RDF/XML for machine-to-machine connection)[17].

Access points are :

1. RBINS: https://darwin.naturalsciences.be/uri/object/[UUID]: e.g. https://darwin.naturalsciences.be/uri/object/79e51258-e6c3-4f7d-9497-d40e243a20e4
2. RMCA: https://darwinweb.africamuseum.be/object/[UUID]: e.g. https://darwinweb.africamuseum.be/object/85005023-5f06-49b4-b2bc-83ec3a65c2c0


https://darwin.naturalsciences.be//uri/data/rdf//feb56ec3-e95f-4578-b51e-ad2525d17075

Aucune information de style ne semble associée à ce fichier XML. L'arbre du document est affiché ci-dessous.

```
−<rdf:RDF>
  −<rdf:Description rdf:about="https://darwin.naturalsciences.be/uri/permalink/data/rdf/feb56ec3-e95f-4578-b51e-ad2525d17075">
     <dc:subject rdf:resource="https://darwin.naturalsciences.be/uri/object/feb56ec3-e95f-4578-b51e-ad2525d17075"/>
    −<dc:creator>
       Simple PHP RDF Script implemented by Felix Hilgerdenaar (BGBM) [PDO adaptation by RMCA]
     </dc:creator>
     <dc:created>2020-12-24T12:46:02+00:00</dc:created>
   </rdf:Description>
  −<rdf:Description rdf:about="https://darwin.naturalsciences.be/uri/object/feb56ec3-e95f-4578-b51e-ad2525d17075">
    −<dc:title>
       CATALOG NUMBER : TAXON : Gyrinus (Gyrinus) caspius Ménétries 1832
     </dc:title>
    −<dc:description>
       CATALOG NUMBER : TAXON : Gyrinus (Gyrinus) caspius Ménétries 1832
     </dc:description>
     <dc:publisher>RBINS</dc:publisher>
    −<dwc:materialSampleID>
       https://darwin.naturalsciences.be/search/view/uuid/feb56ec3-e95f-4578-b51e-ad2525d17075
     </dwc:materialSampleID>
     <dc:modified>2012-10-16 14:39:49.433891</dc:modified>
     <dwc:basisOfRecord>specimen</dwc:basisOfRecord>
     <dwc:scientificName>Gyrinus (Gyrinus) caspius Ménétries 1832</dwc:scientificName>
     <dwc:previousIdentifications>Gyrinus (Gyrinus) caspius Ménétries 1832</dwc:previousIdentifications>
     <dwc:family>Gyrinidae Latreille 1806</dwc:family>
     <dwc:genus>Gyrinus Müller 1764</dwc:genus>
     <dwc:specificEpithet>(Gyrinus)</dwc:specificEpithet>
     <dwc:locality>Belgium ; België ; Belgique ; Belgien</dwc:locality>
   </rdf:Description>
 </rdf:RDF>
```

Figure 23: RDF/XML representation of a specimen accessed from a stable URL

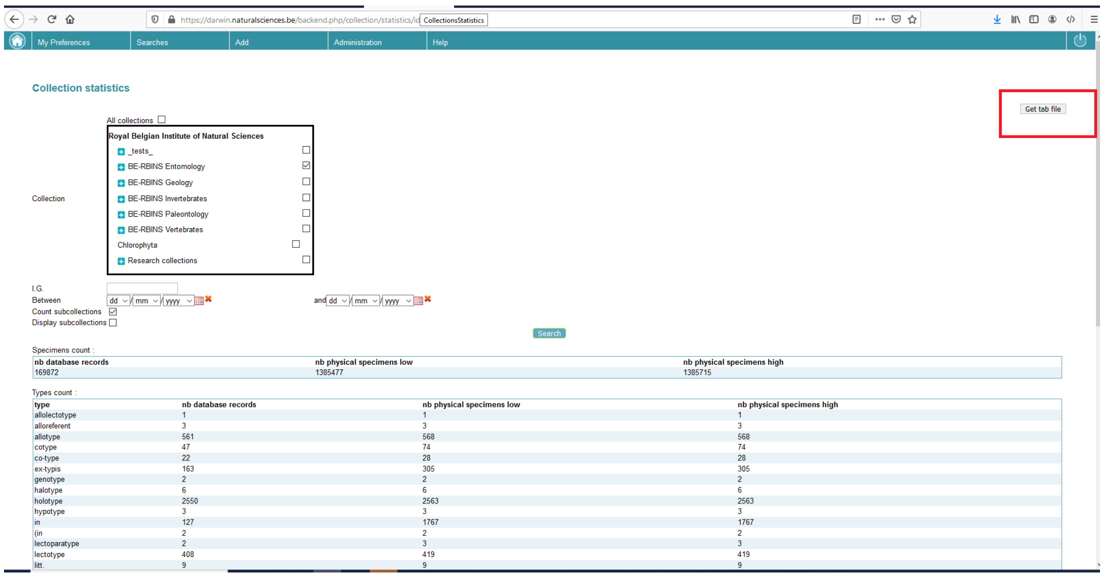
These protocols are actually based on ontological concepts, where locators are defined not only for objects on the web but also for real-world objects [18] and have a certain degree of abstraction. They aim to create knowledge databases whose life expectancy could potentially last beyond the current state-of-arts HTTP and TCP standards; although their current implementation depends on them.

[17] See https://cetafidentifiers.biowikifarm.net/wiki/Main_Page for more details
[18] See also: https://en.wikipedia.org/wiki/HTTPRange-14

4.5 Reworked asynchronous mechanism for reports in tab-delimited format

The reporting mechanism of DaRWIN has been deeply reworked so that it doesn't depend anymore on third-party report servers. Reports are fully embedded in DaRWIN and published as raw CSV outputs. This simplifies considerably the architecture of DaRWIN, making it more self-contained, and therefore, easier to maintain and less demanding in terms of IT infrastructure.
This new report mechanism is based on asynchronous background tasks in PHP. They can be triggered from the web interface, but are actually Bash or DOS console operations (Symfony 1 tasks). A control is made on the type of parameters, to prevent injection of code. A paging and semaphores in AJAX/JSON format allow synchronization of the user web interface with the task. Report requests open new browser tabs with a timer and a "download" button that is displayed once the report is ready.
Reports can export data coming from:

1. saved searches of specimens:
   a PL/PgSQL function, doing dynamic SQL statements has been appended, allowing users to save their search criteria and reproduce the result on demand
2. Taxa and child-taxa (with statics about the number of specimen, the types present in collection and the type of storage)
3. Expeditions
4. I.G. (*Inventaire général*) numbers
5. Collections with details on
   a. The amount of specimens in collections
   b. The amount of types in collections
   c. The taxonomic diversity
   d. The structure of subcollections



Collection statistics

All collections ☐

Collection

Royal Belgian Institute of Natural Sciences
- _tests_ ☐
- BE-RBINS Entomology ☑
- BE-RBINS Geology ☐
- BE-RBINS Invertebrates ☐
- BE-RBINS Paleontology ☐
- BE-RBINS Vertebrates ☐
- Chlorophyta ☐
- Research collections ☐

I.G.
Between dd / mm / yyyy and dd / mm / yyyy
Count subcollections ☑
Display subcollections ☐

Specimens count :

nb database records	nb physical specimens low	nb physical specimens high
169872	1385477	1385715

Types count :

type	nb database records	nb physical specimens low	nb physical specimens high
allolectotype	1	1	1
alloreferent	3	3	3
allotype	561	568	568
cotype	47	74	74
co-type	22	28	28
ex-typis	163	305	305
genotype	2	2	2
halotype	6	6	6
holotype	2550	2563	2563
hypotype	3	3	3
in	127	1767	1767
(in	2	2	2
lectoparatype	2	3	3
lectotype	408	419	419
litt.	9	9	9

Figure 16: Interface to extract collections reports

4.6 Export to DaRWINCore Archives

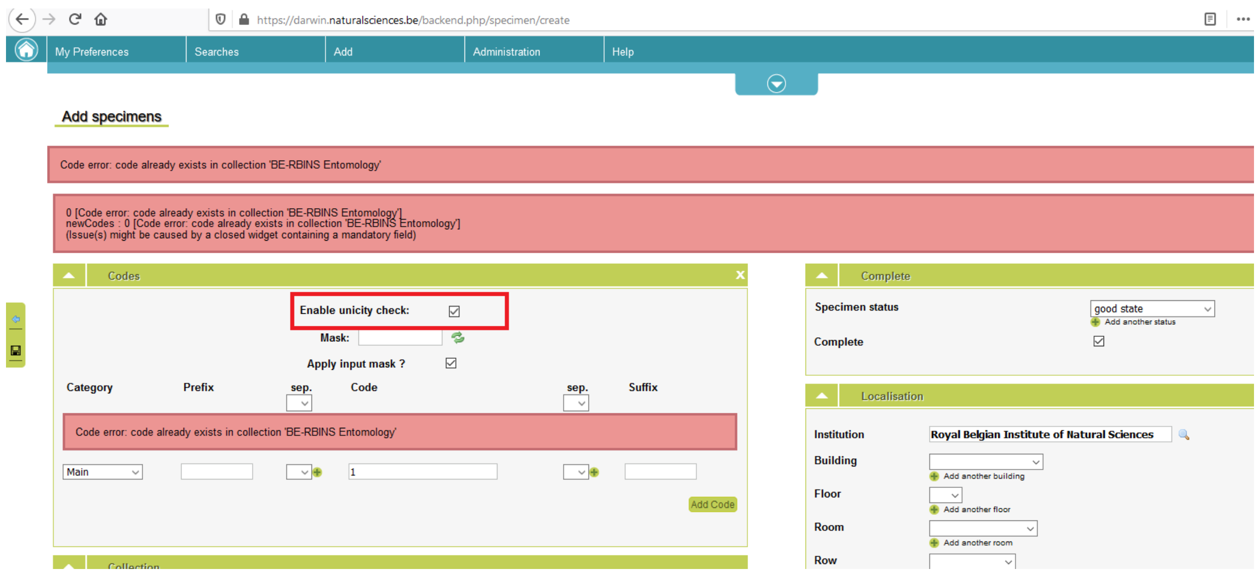
With Thomas Vandenberghe, DaRWIN has also been adapted to extract the data in the two formats used by GBIF. A specific "*gbif*" database schema as been appended, where :

1. Database records can be extracted into DaRWINCore Archive formats (via materialized views)
2. A set or R scripts allows converting collection statistics of DaRWIN into data in EML (Ecological Markup Language), the XML schema for collection metadata

These two data can be mapped afterwards into the GBIF IPT *(Internet Publishing Toolkit)*, the webservice developed by GBIF to provide data to the network. These data are zipped static dumps (DaRWINCore Archive). The same workflow can be used to publish data to the *ElasticSearch* NaturalHeritage portal that can read data in DawinCore Archive format.
This would make DaRWIN data citable, as GBIF associates a DOI to published datasets. The UUID integration described above makes this publishing pipeline more integrated into DaRWIN.
Thomas also improved the normalization of locality description in DaRWIN, adding several with uniqueness constraints tables that can map DaRWIN "tags" (i.e categorized geographical names) to reference authoritative URIs (such as GeoNames, WikiData).
This would ease the integration of DaRWIN data into linked data architecture based on RDF.

4.7 Automatic unique collection numbers and uniqueness check

DaRWIN gives the possibility to link a collection to a sequence of automatically generated collection numbers. This functionality is available both in the manual insertion interface and in batch import from tab-delimited data. The user can also decide whether sub-collections are included in the sequence or not.
The web interface DaRWIN has also a constraint in the web interface preventing the creation of duplicate collection numbers. This constraint is enabled by default but can be disabled via a checkbox (see fig. 25)



Add specimens

Code error: code already exists in collection 'BE-RBINS Entomology'

0 [Code error: code already exists in collection 'BE-RBINS Entomology']
newCodes : 0 [Code error: code already exists in collection 'BE-RBINS Entomology']
(Issue(s) might be caused by a closed widget containing a mandatory field)

Enable unicity check: ✓
Mask:
Apply input mask ? ✓

Category	Prefix	sep.	Code	sep.	Suffix
Main			1		

Code error: code already exists in collection 'BE-RBINS Entomology'

Specimen status: good state
Complete: ✓

Institution: Royal Belgian Institute of Natural Sciences
Building:
Floor:
Room:
Row:

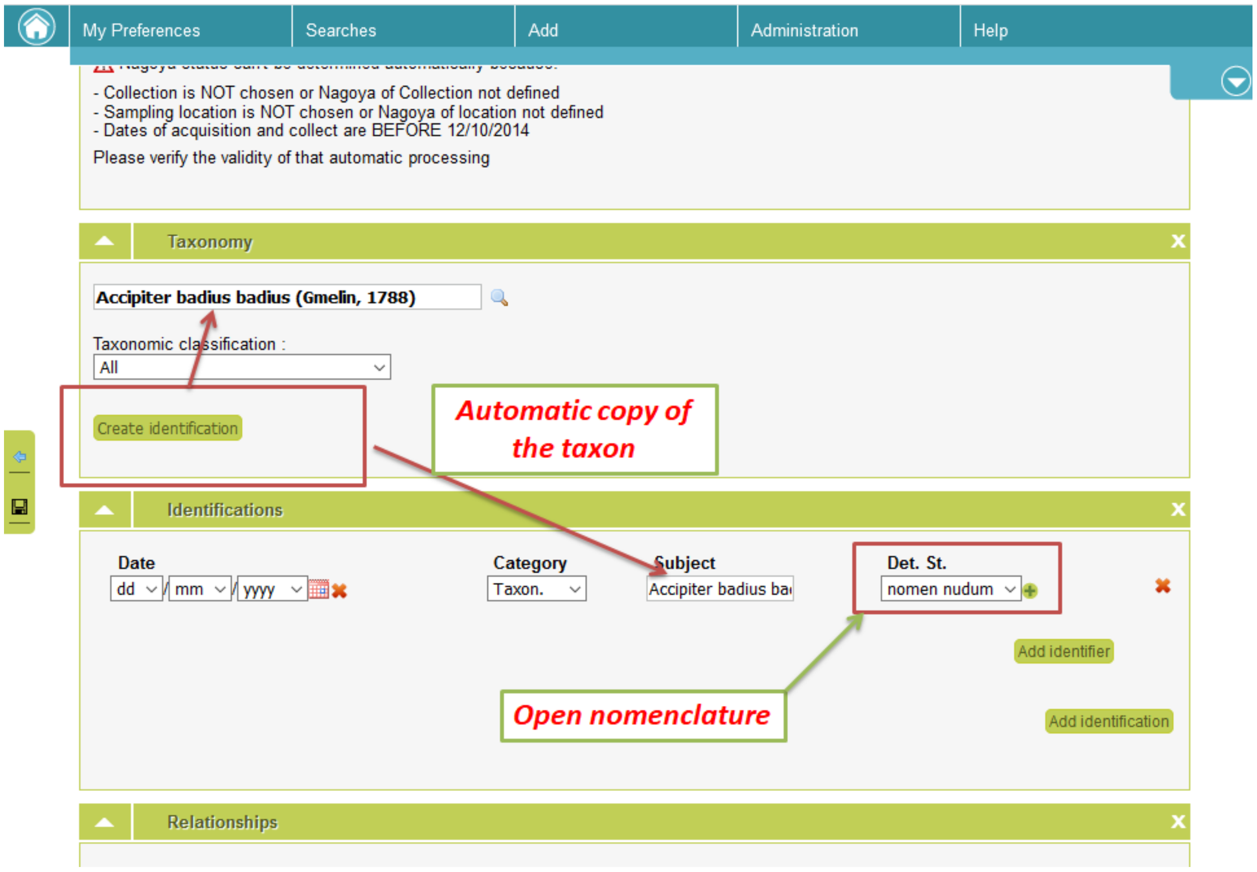
Figure 17: unicity check on collection number

A similar checkbox can enforce or disable the same constraint in batch imports

4.8 Distinction between taxonomic hierarchy and determination events

When analyzing the usage of DaRWIN, we noticed that the "Open nomenclature" information, annotating accuracy of taxonomic identifications[19] were often present in the taxonomy, while they should rather be mentioned in the identification part of DaRWIN. The initial model of DaRWIN was well-designed, allowing to specify a "subject" for each identification operation. A button has been added, which automatically fills this subject with the taxon information. The open nomenclature info can be mentioned via a controlled *determination status* list

[19] https://en.wikipedia.org/wiki/Open_nomenclature



- Collection is NOT chosen or Nagoya of Collection not defined
- Sampling location is NOT chosen or Nagoya of location not defined
- Dates of acquisition and collect are BEFORE 12/10/2014

Please verify the validity of that automatic processing

Accipiter badius badius (Gmelin, 1788)

Taxonomic classification : All

Date	Category	Subject	Det. St.
dd / mm / yyyy	Taxon.	Accipiter badius ba	nomen nudum

Figure 18. Open nomenclature in DaRWIN

4.9 Rotating backup

A backup plan of the database as been introduced, with the following strategy, that was configured via 4 backup (and cleaning) jobs :

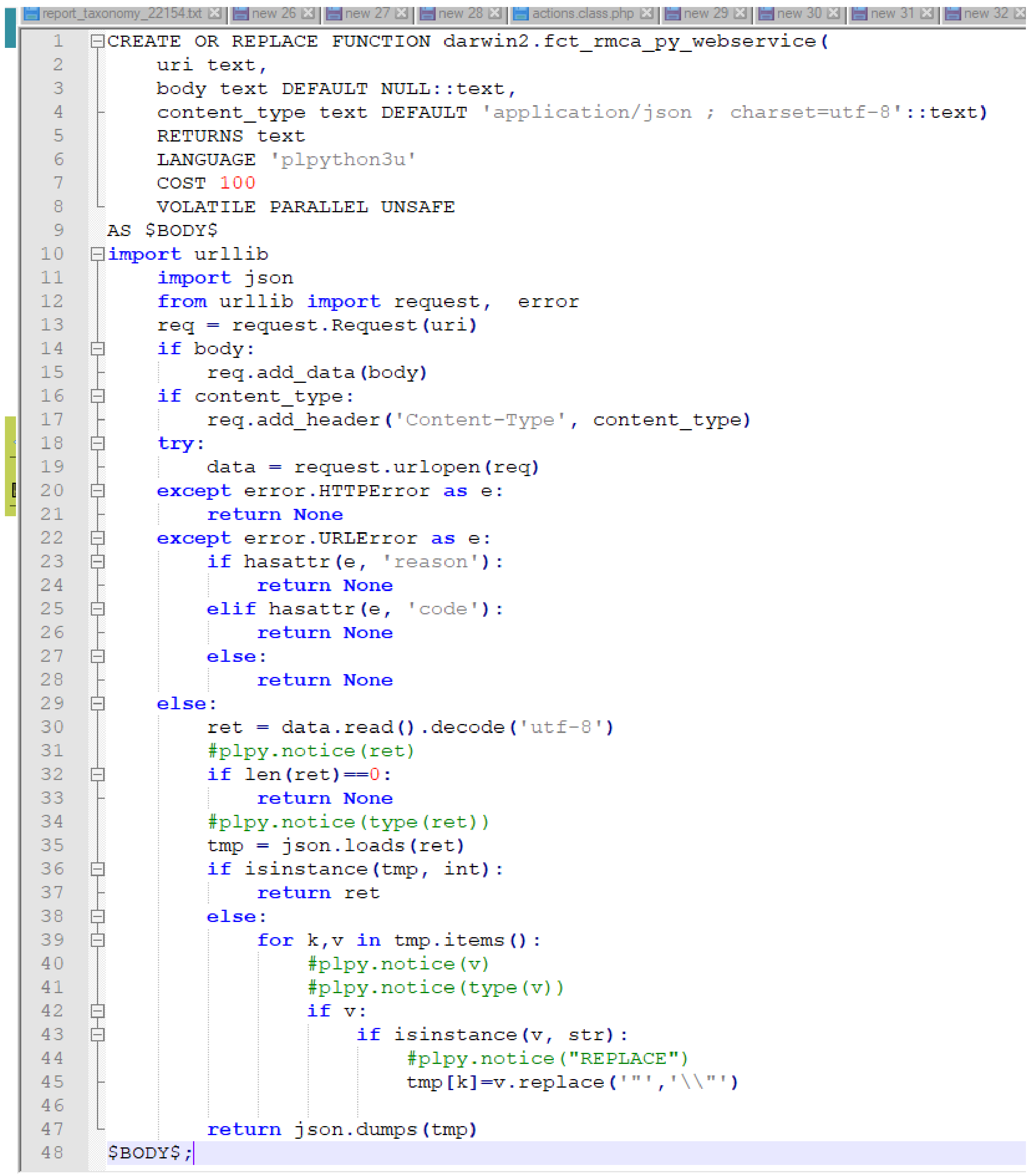
1. a folder for yearly backups, that can be never expunged. It keeps a backup of DaRWIN made each year 24th of December ;
2. a folder for monthly backups, made every first day of the month, keeping only the last 2 files ;
3. a folder for weekly backups, keeping one backup per week (every Monday) and is expunged every Month ;
4. a folder for daily backups, made every night (from Tuesday to Friday). Files older than 4 working days are expunged ;

This way the server only needs to keep 10 files (2 monthly + 4 weekly + 4 daily) at the same time, plus one file per year (but this file could be transferred manually to another server). The whole backup folder (excepting the “yearly” files) weighs between 6 Gigas and 12 Gigas (if DaRWIN contains twice the volume of data as now).

4.10 Embedded PL\PgSQL REST client

Thanks to the Python language module of PostgreSQL, DaRWIN features a REST JSON client that is embedded within the SQL database. The whole result of calls can be directly expressed as a pseudo-table that can be used in JOIN query statements. This allows to compare and clean DaRWIN data with reference databases having a REST API (*GBIF*, *Catalog of Life* for taxonomy, *Overpass Turbo* and *OpenStreetMap/Nominatim* for geographical places etc...) in a very fast and efficient way, as no tedious AJAX or third-party applications or graphical interfaces are needed anymore (if the user is familiar with SQL). Calls to remote servers can be done by using only SQL language. This feature has been tested by case-study of geo-referencing, to find the geo-referenced paths (lines expressed as Well Known Text polygons) of present in DaRWIN in a semi-automated way by using *OverPass Turbo* and *Nominatim* APIs[20], by using their name and country as search criteria. Results were promising, however a graphical interface would still need to be developed to transform this into a front-end application for non-IT users.

[20] https://nominatim.org/release-docs/latest/api/Search/ (query engine and search interface for OpenStreetMap data)



```
CREATE OR REPLACE FUNCTION darwin2.fct_rmca_py_webservice(
    uri text,
    body text DEFAULT NULL::text,
    content_type text DEFAULT 'application/json ; charset=utf-8'::text)
    RETURNS text
    LANGUAGE 'plpython3u'
    COST 100
    VOLATILE PARALLEL UNSAFE
AS $BODY$
import urllib
    import json
    from urllib import request,  error
    req = request.Request(uri)
    if body:
        req.add_data(body)
    if content_type:
        req.add_header('Content-Type', content_type)
    try:
        data = request.urlopen(req)
    except error.HTTPError as e:
        return None
    except error.URLError as e:
        if hasattr(e, 'reason'):
            return None
        elif hasattr(e, 'code'):
            return None
        else:
            return None
    else:
        ret = data.read().decode('utf-8')
        #plpy.notice(ret)
        if len(ret)==0:
            return None
        #plpy.notice(type(ret))
        tmp = json.loads(ret)
        if isinstance(tmp, int):
            return ret
        else:
            for k,v in tmp.items():
                #plpy.notice(v)
                #plpy.notice(type(v))
                if v:
                    if isinstance(v, str):
                        #plpy.notice("REPLACE")
                        tmp[k]=v.replace('"','\\"')

        return json.dumps(tmp)
$BODY$;
```

Figure 19. Code of the SQL/Python REST client

4.11 Introduction of identifiers for linked data

Identifiers protocols have been introduced in DaRWIN, such as :

a. ORCID for peoples
b. GrSCICOll for collections
c. GRID for institutions
d. DOI for bibliographical reference

These identifiers can be used as search criteria to find related resources. They are also resolvable (DaRWIN provided a link to the related resource available on the web when available). New protocols can be dynamically added, as controlled lists and keyword dictionaries in DaRWIN are not static, but can be modified and updated from the interfaces

4.12 Autocomplete fields

Autocomplete fields have been appended in the search interface of DaRWIN, to guide the user when he introduces search criteria and avoid typos and misspellings. This concern (among others) the collections numbers and the names of expeditions

4.13 Mass actions

Additional mass actions (allowing batch update of specimens) have been programmed, by using the existing code template. This concerns :

1. Adding a locality information to several specimens
2. Adding a property to specimens
3. Adding or modifying the sampling data of specimens
4. Changing the Nagoya status of specimens

4.14 Better Navigability

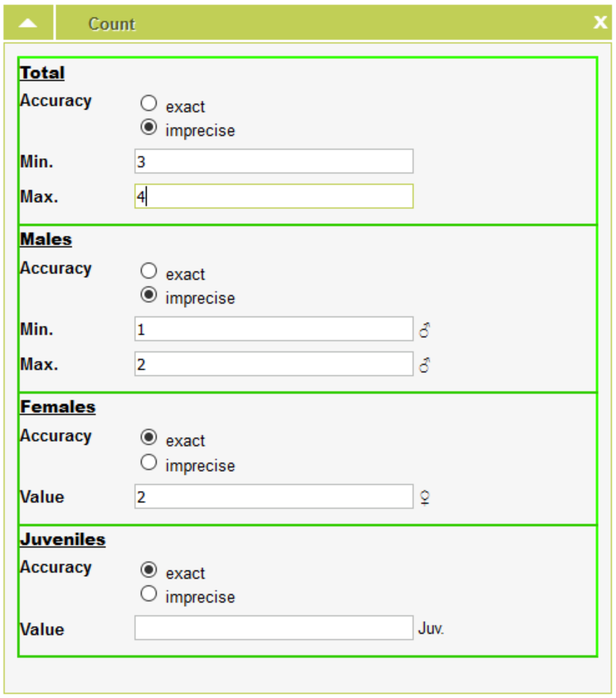
Links have been appended to navigate from the list of I.G. numbers (which identify expeditions dispatched among several collections or sub collections) to :

1. The list of related specimens
2. The list of related taxa
3. The list of related collecting places
4. The list of related collectors and expeditions

These I.G.can be exported into a report in PDF format at the RMCA

4.15 Count of specimens

The Widget “Count” in specimen edit form has been improved to be more readable. Columns have also been added in the results of a search



Count

Total

Accuracy	exact [✓ imprecise]
Min.	3
Max.	4

Males

Accuracy	exact [✓ imprecise]
Min.	1 ♂
Max.	2 ♂

Females

Accuracy	[✓ exact] imprecise
Value	2 ♀

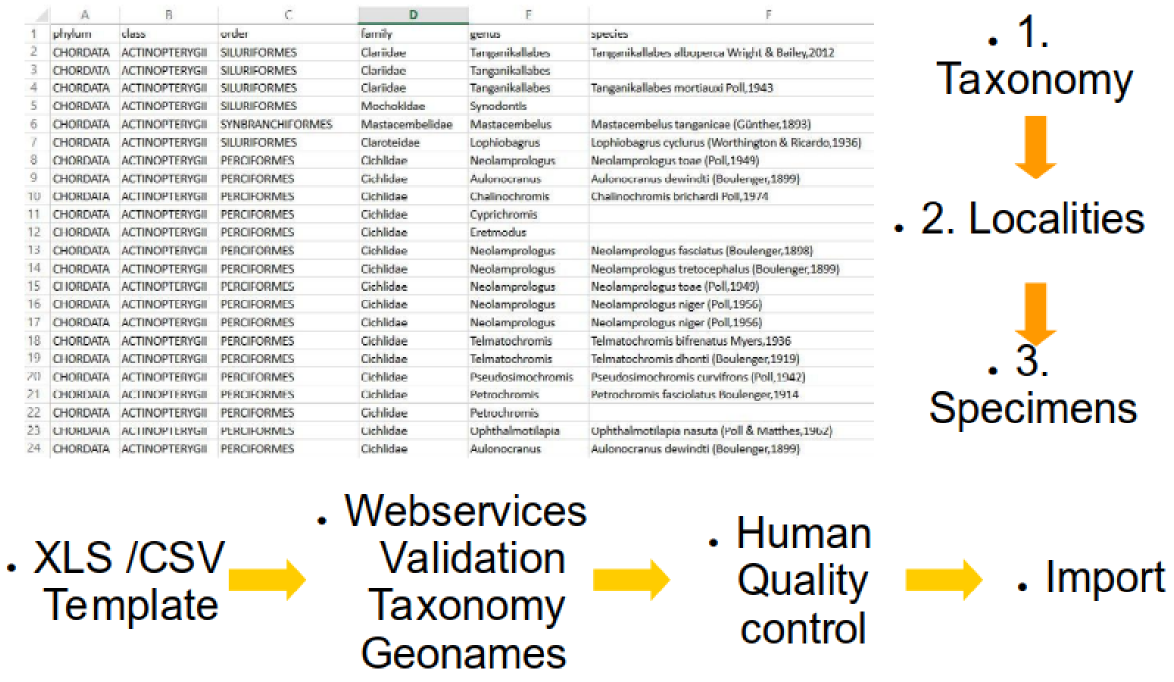
Juveniles

Accuracy	[✓ exact] imprecise
Value	Juv.

Figure 20: The new Count widget

4.16 New Import system using several XLS templates

A new import module was developed. It is based on XLS (or any other spreadsheet able to generate tab-delimtied documents) templates and a 3 steps import: the complete description of the new module is available in the annex 4 - BBR/175/A3 NaturalHeritage: The Import module of the DaRWIN Collection Management System



	A	B	C	D	F	F
1	phylum	class	order	family	genus	species
2	CHORDATA	ACTINOPTERYGII	SILURIFORMES	Clariidae	Tanganikallabes	Tanganikallabes alboperca Wright & Bailey,2012
3	CHORDATA	ACTINOPTERYGII	SILURIFORMES	Clariidae	Tanganikallabes	
4	CHORDATA	ACTINOPTERYGII	SILURIFORMES	Clariidae	Tanganikallabes	Tanganikallabes mortiauxi Poll,1943
5	CHORDATA	ACTINOPTERYGII	SILURIFORMES	Mochokidae	Synodontis	
6	CHORDATA	ACTINOPTERYGII	SYNBRANCHIFORMES	Mastacembelidae	Mastacembelus	Mastacembelus tanganicae (Günther,1893)
7	CHORDATA	ACTINOPTERYGII	SILURIFORMES	Claroteidae	Lophiobagrus	Lophiobagrus cyclurus (Worthington & Ricardo,1936)
8	CHORDATA	ACTINOPTERYGII	PERCIFORMES	Cichlidae	Neolamprologus	Neolamprologus toae (Poll,1949)
9	CHORDATA	ACTINOPTERYGII	PERCIFORMES	Cichlidae	Aulonocranus	Aulonocranus dewindti (Boulenger,1899)
10	CHORDATA	ACTINOPTERYGII	PERCIFORMES	Cichlidae	Chalinochromis	Chalinochromis brichardi Poll,1974
11	CHORDATA	ACTINOPTERYGII	PERCIFORMES	Cichlidae	Cyprichromis	
12	CHORDATA	ACTINOPTERYGII	PERCIFORMES	Cichlidae	Eretmodus	
13	CHORDATA	ACTINOPTERYGII	PERCIFORMES	Cichlidae	Neolamprologus	Neolamprologus fasciatus (Boulenger,1898)
14	CHORDATA	ACTINOPTERYGII	PERCIFORMES	Cichlidae	Neolamprologus	Neolamprologus tretocephalus (Boulenger,1899)
15	CHORDATA	ACTINOPTERYGII	PERCIFORMES	Cichlidae	Neolamprologus	Neolamprologus toae (Poll,1949)
16	CHORDATA	ACTINOPTERYGII	PERCIFORMES	Cichlidae	Neolamprologus	Neolamprologus niger (Poll,1956)
17	CHORDATA	ACTINOPTERYGII	PERCIFORMES	Cichlidae	Neolamprologus	Neolamprologus niger (Poll,1956)
18	CHORDATA	ACTINOPTERYGII	PERCIFORMES	Cichlidae	Telmatochromis	Telmatochromis bifrenatus Myers,1936
19	CHORDATA	ACTINOPTERYGII	PERCIFORMES	Cichlidae	Telmatochromis	Telmatochromis dhonti (Boulenger,1919)
20	CHORDATA	ACTINOPTERYGII	PERCIFORMES	Cichlidae	Pseudosimochromis	Pseudosimochromis curvifrons (Poll,1942)
21	CHORDATA	ACTINOPTERYGII	PERCIFORMES	Cichlidae	Petrochromis	Petrochromis fasciolatus Boulenger,1914
22	CHORDATA	ACTINOPTERYGII	PERCIFORMES	Cichlidae	Petrochromis	
23	CHORDATA	ACTINOPTERYGII	PERCIFORMES	Cichlidae	Ophthalmotilapia	Ophthalmotilapia nasuta (Poll & Matthes,1962)
24	CHORDATA	ACTINOPTERYGII	PERCIFORMES	Cichlidae	Aulonocranus	Aulonocranus dewindti (Boulenger,1899)

Figure 21: The new import procedure in 3 steps using XLS templates

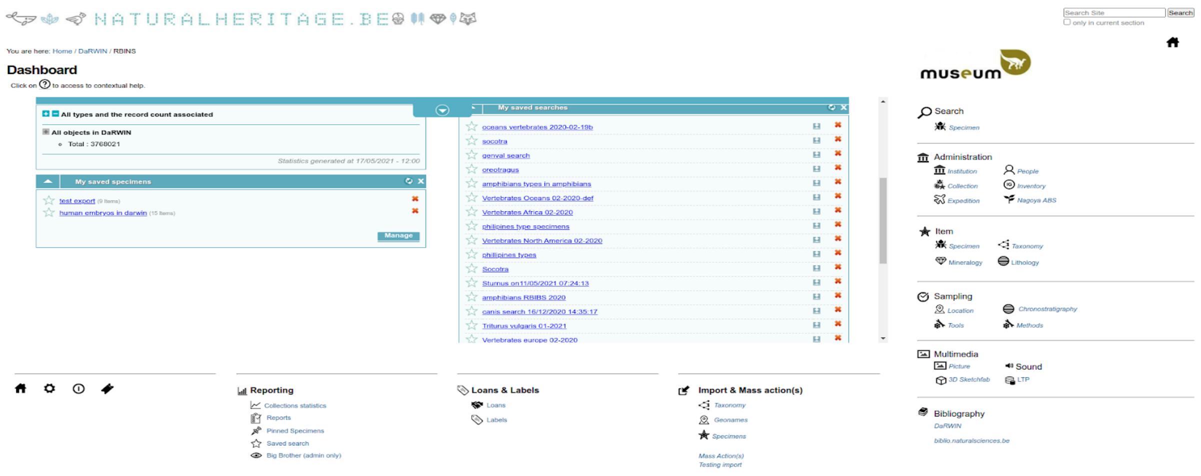
4.17 New module for labels creation, export and printing

A new module for the labels was developed. It allows direct printing with specific printer drivers or indirect printing using CSV file): the complete description of the new features is available in annex 2 - BBR/175/A3 NaturalHeritage: The Label module of the DaRWIN Collection Management System

4.18 New module for the loans.

A loan module was previously developed at RBIN, but a new one was developed at RMCA in the framework of Natural Heritage. The complete description of the new features is available in annex 3 - BBR/175/A3 NaturalHeritage: The loan module of the DaRWIN Collection Management System
4.19 New graphic user interface (DaRWIN backend) with integration in the institution website(s)

It was designed an alternated user interface for the DaRWIN CMS which can be embedded in the Institutional website: the complete description of the new interface is available in the annexe 6 - BBR/175/A3 NaturalHeritage: The new interface of the DaRWIN Collection Management System.



You are here: Home / DaRWIN / RBINS

Dashboard

Click on ? to access to contextual help.

All types and the record count associated

All objects in DaRWIN
- Total : 3768021

Statistics generated at 17/05/2021 - 12:00

My saved specimens

- test export (9 items)
- human embryos in darwin (15 items)

My saved searches

- oceans vertebrates 2020-02-19b
- socotra
- genval search
- oreotragus
- amphibians types in amphibians
- Vertebrates Oceans 02-2020-def
- Vertebrates Africa 02-2020
- philipines type specimens
- Vertebrates North America 02-2020
- phillipines types
- Socotra
- Sturnus on11/05/2021 07:24:13
- amphibians RBIBS 2020
- canis search 16/12/2020 14:35:17
- Triturus vulgaris 01-2021
- Vertebrates europe 02-2020

Figure 22: The new interface of DarWIN CMS integrated in the NH Portal

5. DISSEMINATION AND VALORISATION

The DaRWIN CMS is now a mature and powerful collection management system.
As it is a solution developed in Open Source technologies, it allows a transfer to new "clients" / "partners" in Belgium in the framework of the DiSSCo infrastructure development or abroad.
The system has been presented to Frederic Francis, Dean of the faculty of Biology at the University of Liège, in September 2020, as a possible collection data management system for the Zoological and paleontological collections of the University of Liège.
The size of the Uliège University collection is estimated to be more than 5 millions of specimens and is the third collection Belgium after RBINS and RMCA.

The geographic module and the functionalities allowing translation of geographic names from OSM data have been presented to the DISSCO community during a webinar in June 2020 and drew the attention of IT engineers of GBIF, who also used bottom-up approaches or controlled vocabularies
DaRWIN has also been presented to the IT managers of the Herbarium of the Natural Museum of Geneva, who mentioned that the mechanism to integrate, clean and reconcile external value was a unique asset of the system.
DaRWIN has also been presented in April 2019 in Tervuren and Kigali (Rwanda) as a possible web portal for a biological project set up by the University Rwanda and the JRS Biodiversity Foundation, focusing on the ecological threats and biodiversity loss in the Mukungwa river area.
DaRWIN has also been used to integrate recent ichthyological data from DGD *MBISA* and *BICS* projects in 2020, following a workshop organized with the Professeur Carl Hopkins from Cornell University at the RMCA in February 2020. DaRWIN is used to store specimen description along with graphs and EODs (*Electric Organic Discharges*) data from Mormyiridae collected by scientists from 6 Universities of Central Africa (U*niversity of Lubumbashi, University of Kisangani, Université Officielle de Bukavu, University of Mbanza-Ngungu, Université Marien Ngouabi of Brazzaville, University of*

Bujumbura). There are also ongoing discussions within the *MBISA* project to use DaRWIN to store and keep EOD data from the University of Cornell.

6. PUBLICATIONS

Adam M, Theeten F, Herpers J-M, Vandenberghe T, Semal P, Van den Spiegel D, Duchesne P-A (2019) DaRWIN: An open source natural history collections data management system. Biodiversity Information Science and Standards 3: e39054. https://doi.org/10.3897/biss.3.39054
(Poster in conference)

7. ACKNOWLEDGEMENTS

We acknowledge the role of Paul-André Duchesnes, the inceptor of DaRWIN. Many of the feature he implemented in DaRWIN are visionary, like the functional staging interface for the integration of external data and the detection of duplicates in auxiliary tables, the replacement of foreign keys by dynamic relations connecting several tables, the systematic log of historical data and the idea to create a system which is halfway between a relational database and a data warehouse. The data model was sound and he and his team made an impressive work on the qualitative and quantitative point of views. Paul-André was present to guide us through the system and to organize a smooth technical transition.
Max Fernandez and Pascale Lahogue from the Geological service of the RMCA also helped us to understand the difference and complementarity between chronostratigraphical classification and lithostratigraphical classifications and helped us to develop the related importation template.

Authors: Franck Theeten, Jean-Marc Herpers, Didier VandenSpiegel & Patrick Semal