

## ObServe: Database and operational software for longline and purse seine fishery data

Pascal Cauquil<sup>1\*</sup>, Njaratiana Rabearisoa<sup>2</sup>, Philippe S. Sabarros<sup>1,2</sup>, Pierre Chavance<sup>1</sup> and Pascal Bach<sup>1</sup>

<sup>1</sup>Institut de Recherche pour le Développement (IRD), Observatoire Thonier (OT), UMR 248 MARine Biodiversity, Exploitation & Conservation (MARBEC), Centre de Recherche Halieutique Méditerranéenne et Tropicale (CRHMT), Avenue Jean Monnet, BP 171, CS30171, 34203 Sète Cedex, France

<sup>2</sup>Institut de Recherche pour le Développement (IRD), Observatoire Thonier (OT), UMR 248 MARine Biodiversity, Exploitation & Conservation (MARBEC), Hydrô Réunion, Magasin n°10, Darse de pêche hauturière, Port Ouest, 97420 Le Port, La Réunion, France

\*corresponding author, email: [pascal.cauquil@ird.fr](mailto:pascal.cauquil@ird.fr)

**ABSTRACT.** Observation data collected aboard fishing vessels are essential to describe the impact of fisheries on fish community. The *Institut de Recherche pour le Développement* (IRD, France) has been sending observers aboard tropical purse-seiners since 1995 in the Atlantic and Indian Oceans, and longliners since 2007 in the Indian Ocean. Since 2005, IRD is appointed by the European Union (EU) and the French *Direction des Pêches Maritimes et de l'Aquaculture* (DPMA, French government) to conduct scientific observations aboard French vessels to monitor tropical fisheries in the framework of EU *Data Collection Framework* (DCF). To monitor this program, the *Observatoire Thonier* (OT) from IRD has been developing since 2010 an information system named *ObServe* that is intended to manage data collected in the framework of DCF. *ObServe* consists of (i) a central database based on *PostgreSQL*, (ii) a *Java*-based software used for data acquisition and management, and (iii) data synchronization features between these two modules. This system was originally dedicated to purse-seine fisheries but now supports longline ones. Furthermore, historical data were transferred into *ObServe* that now covers 1995-2015 for purse-seine observer data and 2002-2015 for the pelagic longline. *ObServe* was designed to be generic in order to cover the needs of any observation program of purse-seine or longline fisheries. *ObServe* is being used by several institutes, administration and companies that carry out observer surveys on purse-seiners: IEO and AZTI (Spain), SFA (Seychelles), Oceanic Development (France), BiGeye (Ivory Coast), *Terres Australes et Antarctiques Françaises* (France), and longliners: Hydrô Run (Reunion Island, France) and *Le Parc Naturel Marin de Mayotte* (France). *ObServe* is an open source system and can be easily adopted by other partners that have to manage observer data for purse-seine and longline fisheries.

**KEYWORDS.** Fishery | Pelagic longline | Purse-seine | Observation | Information system | Database | Software | UML | PostgreSQL | Java

## 1. Introduction

Information collected at-sea by observers regarding both fishing practice and captures is one of the pillar of the Ecosystem Approach to Fisheries (EAF). Data on whether fish are caught, when, where and how, inform a wide variety of tasks for fishery management. Such data are particularly useful to assess the state of targeted species as well as those that are accidentally caught. Many fish stocks are being exploited close to their maximum sustainable yield (MSY) and plenty of other stocks are over-exploited, needing therefore a cautious monitoring of their recovery. For instance, high resolution data on fishing operations and captures by species reported at the vessel level are used for stock assessment purposes. Actually, this high resolution level is needed to detect drivers of the high frequency variability of catches. Resource abundance indices are inferred in space and time from these data. Also, within the general context of at-sea monitoring, surveillance and control (MSC), such collected information provide individual vessel data to verify whether fishing operations were realized according to compliance rules.

Since 2001, the Common Fisheries Policy (CFP) dedicated funds to help national authorities collect both economic and biological data about all aspects of fisheries management. In 2001 started the Data Collection Regulation (DCR) in order to coordinate collection and management of fisheries data within the EU. In 2008, the Data Collection Framework (DCF) replaced the DCR. DCF was undertaken to extent the DCR scope to the management and the use of data, adopting an ecosystem approach, better rules on the access and use of data, broader data coverage and stronger regional coordination.

Since 2005, the *Observatoire Thonier* (OT) from *Institut de Recherche pour le Développement* (IRD) is appointed by the *Direction des Pêches Maritimes et de l'Aquaculture* (DPMA, French government) under the EU DCF to conduct scientific observations aboard French flag vessels targeting large pelagic fishes: purse seiners operating in the Atlantic and Indian Oceans, and longliners based in Reunion Island and Mayotte that operate in the South West Indian Ocean (French and Malagasy EEZs). To carry out this work, OT has been developing since 2010 an information system, *ObServe*, intended to manage observation data collected on purse seiners. With this technological experience on one hand, and experience in longline fishery data collection since 1995 on the other hand, *ObServe* was improved in 2015 to carry out the management of observation data in longline fisheries. The system has been implemented based on the regulatory objectives of IRD but is intended to be generic as to manage any type of monitored longline trips, with various levels of precision in the data. *Observe* consists of a central database, a software for data acquisition and management that can be used offline, and synchronization features between the two.

In this article we present the data model, the data acquisition and management software, as well as the current contents of the database and some examples of automated reporting produced for longline data.

## 2. Data model description

### 2.1. Shared and specific tables for purse-seine and longline in the data model

When it was decided to integrate the management of longline data to *ObServe*, the approach was to develop an independent data model that met the expectations of longline data, and then to compare this model to the model used for purse-seine data in *ObServe* in order to identify the tables and references that could be shared by the two fishery types.

Regarding reference tables, the purse seine model has 35 tables while the longline model requires 46 tables, of which 21 can be shared in order to avoid redundancy and to facilitate data management (e.g., countries, species, vessels, etc.). Concerning tables containing the observation data, we are here dealing with fine-scale and fishery-specific data, and it seemed detrimental to share tables as this could deserve legibility and ease of data exploitation.

Finally, tables were allocated to three *SQL* schemas:

- the *observe\_common* schema contains shared references data tables (21 tables),
- the *observe\_seine* schema contains references and purse seine specific tables (47 tables),
- the *observe\_longline* schema contains references and longline specific tables (34 tables).

## **2.2. Synchronization between multiple databases**

*ObServe* consists of a central database and multiple instances of the acquisition and management software. The software can be used offline (without connection to the central database) as it uses a built-in instance of the database model that is identical to the central database model. Synchronization functions are used to (i) download/update reference data from the central database to the software instance, and (ii) upload data collected (and saved offline) by the observer into the central database.

To enable these functions, each record is identified by two unique identifiers:

- The primary key is an alphanumeric field, named *topiaid*, that is composed of a radical, the *UNIX* date of creation in milliseconds, and a random suffix. This structure serves as a unique identifier (like a UUID),
- The second identifier, *topiaversion*, contains the version number of the record. It is set to 0 at the creation of the record (when the *topiaid* is generated) and is incremented whenever the record is updated.

Thereby, in presence of two trips with the same identifier, one in the central database, the other one in the built-in database of the acquisition software, the system will ask the user which one is to be kept in order to prevent duplications. The combination of *topiaid* and *topiaversion* is especially useful to determine updates of reference data in local databases when references are out-of-date.

## **2.3. Data model for longline data**

Considering data collected in the framework of EU DCF and data collected during scientific surveys, there are 3 levels of observation, from the less to the more detailed observations:

- commercial self-reported trips: these consist of data collected by the fishing boat crew, generally the Captain. The requested data are designed to be easy to record and compatible with the workload of fishermen during fishing operations. Captures are recorded by lot of species or group of species at the level of the fishing operation as it is in logbooks. The difference with official logbook data is that self-reported data contains additional information on bycatch and depredation (Bach et al., 2013; Sabarros et al., 2013).

- commercial observed trips: these are commercial trips monitored by a scientific observer devoted to this task. The main difference with the self-reported data lies in the ability to model the longline schema, and that captures are here recorded individually and can be located on the – beforehand defined – longline schema (Bach et al., 2014).
- scientific trips: those trips are mainly carried out by scientists for scientific purposes only using experimental longlines equipped with temperature depth recorders and hook timers (aboard professional or scientific longliners). They are set up and observed by scientists themselves. Informations collected on captures include the time of catch and the exact location of the catch on the longline (hook). They are the most detailed trips.

A set of forms was designed for each type of trip to collect data at-sea (Bach et al., 2009). These forms were used as a starting point for the development of the data model, which main characteristic is its ability to manage these different observation scales.

The reading entry point of the model is the *Trip* entity. This entity is associated, either directly or through other entities, to all the other observation tables of *observe\_longline* schema. The model is to be read in a hierarchical way, from the most general level (a trip) to the finest level (a characteristic of a catch for example), through the description of activities (entity *Activity*) and fishing operations (entity *Set*). Other entities gravitate around *Set* to allow global definition of the fishing operation (entities *FloatlinesComposition*, *BranchlinesComposition*, *HooksComposition*, *BaitsComposition*, *LightsticksColor*, *LightsticksType*) and full schema modeling of the longline setup (entities *Section*, *Basket*, *Branchline*). We present here the different entities of the data model through several views of the UML class diagram.

### 2.3.1. Trip view

The entity *Trip* (Fig. 1) is at the top of the database model hierarchy. A trip can be identified with a unique identifier, *topiaid*, for technical use, and also by a vessel code and a departure date for a practical use. The description of a trip includes its type (i.e. self-reported, commercial observed or scientific) and various actors (captain, observer and data entry operator) identifiers. A trip may include 0..\* activities.

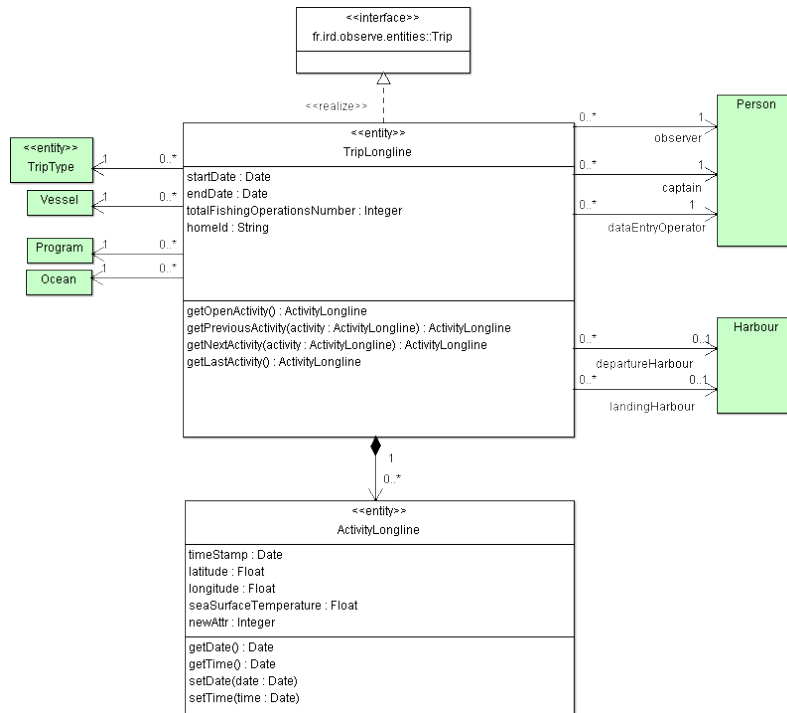


Figure 1. Trip longline view.

2.3.2. Activity view

This view illustrates an observation, characterized by a type of observation (*VesselActivity*), a time stamp and a geographical position (Fig. 2). It may include the description of encounters with protected species at sea (*Encounter*), sensors (*SensorUsed*) and declared fishing zones (*FpaZone*). If the activity type is different from a fishing operation, description stops at this level. If the activity type is a fishing operation, its description continues with the *Set* entity presented in the next section. An activity consists of 0..1 fishing operation.

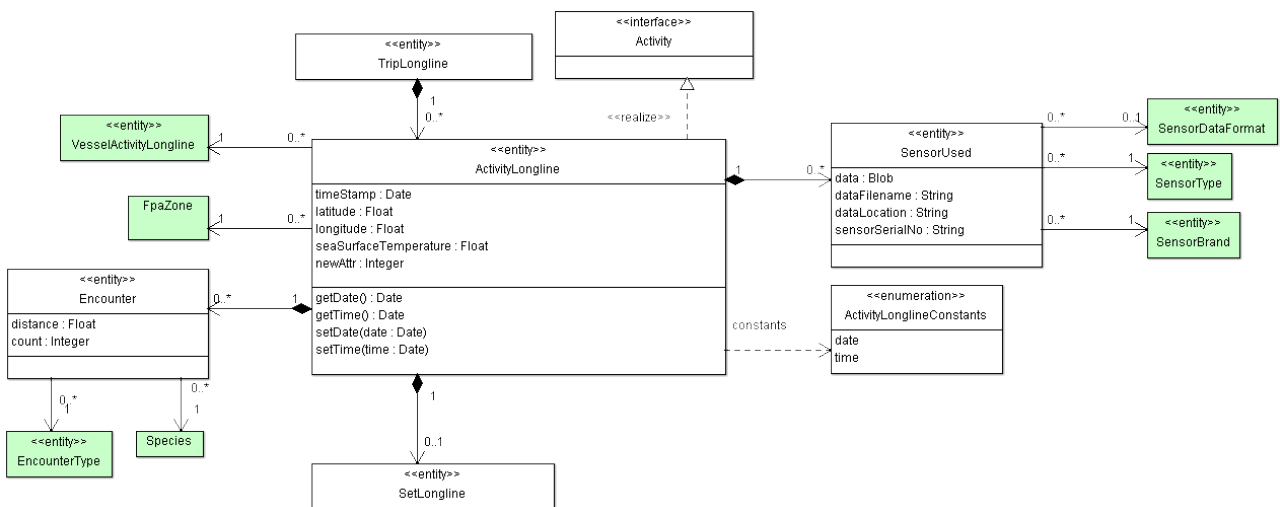


Figure 2. Activity longline view.

2.3.3. Set view

The *Set* view is presented in Figure 3. A set is a particular activity that extends in time and space in the form of a spatial bounding box. In this case, spatio-temporal properties of the *Activity* entity are documented in the *Set* entity by time stamps and positions of longline setting and hauling (*Set.SettingStartTimestamp*, *Set.SettingStartLatitude*, *Set.SettingStartLongitude*, *Set.SettingEndTimestamp*, *Set.SettingEndLatitude*, *Set.SettingEndLongitude*). The *Set* entity contains parameters of the fishing operation setup: use of shooter, swivels and lightsticks characteristics, shape of the line setting, target depth, direction of hauling, line breaks, etc. Entities such as *BranchlinesComposition*, *FloatlinesComposition*, *HooksComposition* and *BaitsComposition* describe the global composition and structure of the set. *BranchlinesPerBasketCount*, *BasketsPerSectionCount*, and *TotalSectionsCount* entities quantify the number of sections, baskets and branchlines, and are used as a starting point to define the schema of the longline setup.

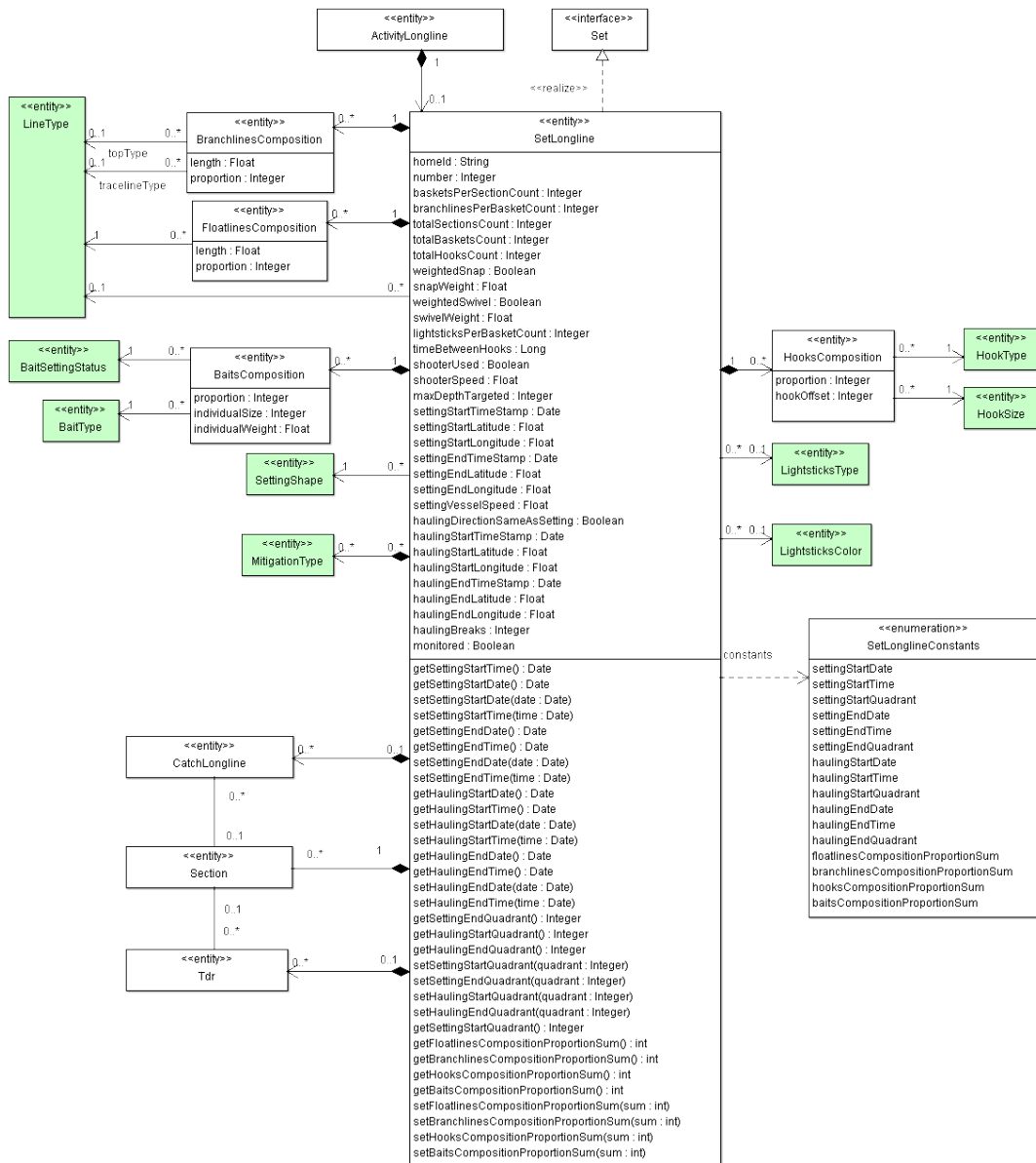


Figure 3. Set view (for longline).

2.3.4. Schema view

The schema of the longline setup is supported by the *Section*, *Basket* and *Branchline* entities. The detailed definition of the longline schema implies specifying for each section, each basket and each branchline, how the longline is structured, with details of the constitution of each branchline when possible: length and material of its upper and lower parts, swivel characteristics, bait type and its status at setting and hauling, hook characteristics, hook timer position and data, branchline and hook status at hauling.

This task is facilitated by the general user interface (GUI) that uses the *Set.BranchlinesPerBasketCount*, *Set.BasketsPerSectionCount* and *Set.TotalSectionsCount* properties to generate a basic longline setup. The data entry operator can then add/remove/move elements to achieve the exact definition of the longline setup.

The definition of the longline schema allows to accurately locate captures on the longline, as well as deprecation events and TDR deployments. The definition of detailed longline schema is not mandatory because the required information are not always available. This point among others shows how the data model deals with different levels of details in collected data. The longline schema is presented as three UML views respectively corresponding to the *Section*, *Basket* and *Branchline* entities (Figs. 4-6).

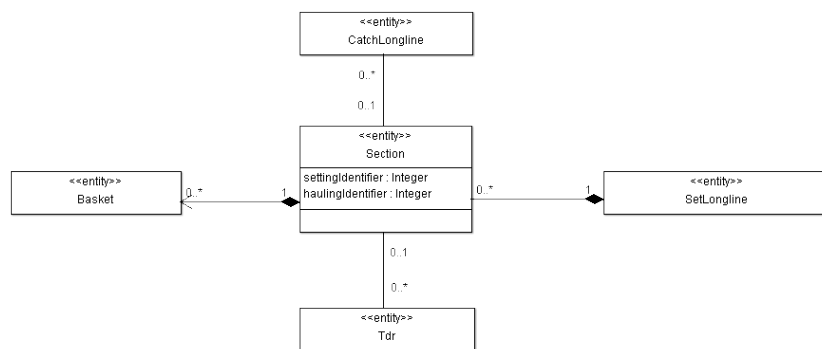


Figure 4. Section view

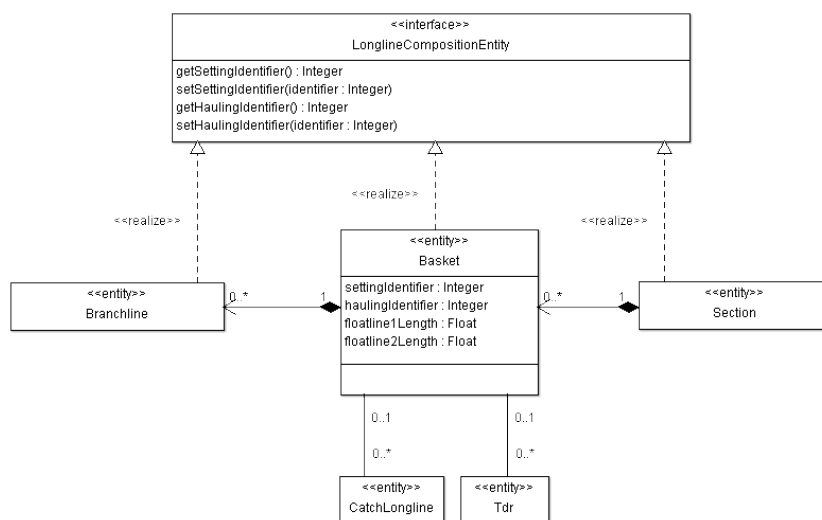


Figure 5. Basket view.

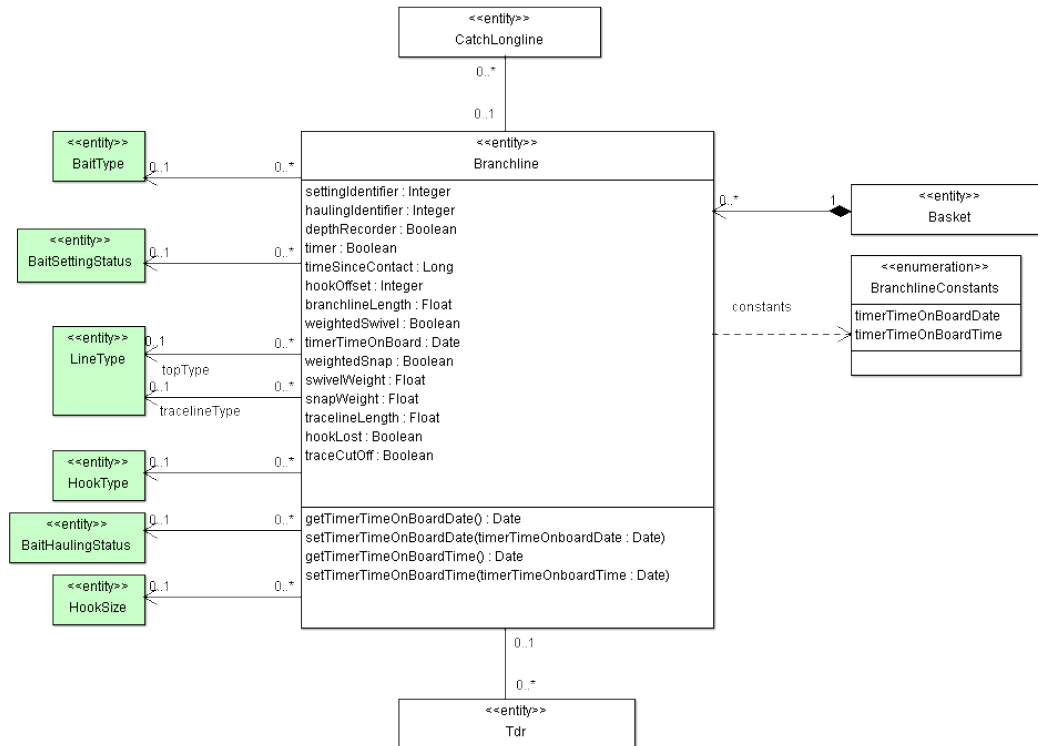


Figure 6. Branchline view.

### 2.3.5. Catch view

The *Catch* view (Fig. 7) illustrates the entities used to describe each capture:

- species,
- number and total weight (if entered by lot),
- capture status, and if necessary, discard status,
- hook position (if known),
- fate,
- length and weight measures,
- depredation,
- stomach content and sexual maturity.

More precisely, captures can be recorded by individual or by lot of individuals. In the first case the property *Catch.count* (number of individuals) is invariably set to 1 by the GUI, and the *Catch.totalWeight* (total weight of lot) property is disabled. In the second case, the *Set.HookPosition* (position of the hook on the catch) property is disabled because not relevant for a lot. *Catch.count* and *Catch.totalWeight* properties are enabled in that case.

Another point distinguishes catch recorded by lot from those individually recorded: catch recorded by lot cannot be allocated to a particular branchline, basket or section. On the contrary, when recording an individual capture, it is possible to locate the catch on the longline: the user has access to three additional



drop-down lists that list the identifiers of sections, baskets, and branchlines. Those identifiers are provided by the detailed longline schema that has been previously defined. In the database, this is achieved by adding the corresponding identifiers on the catch record.

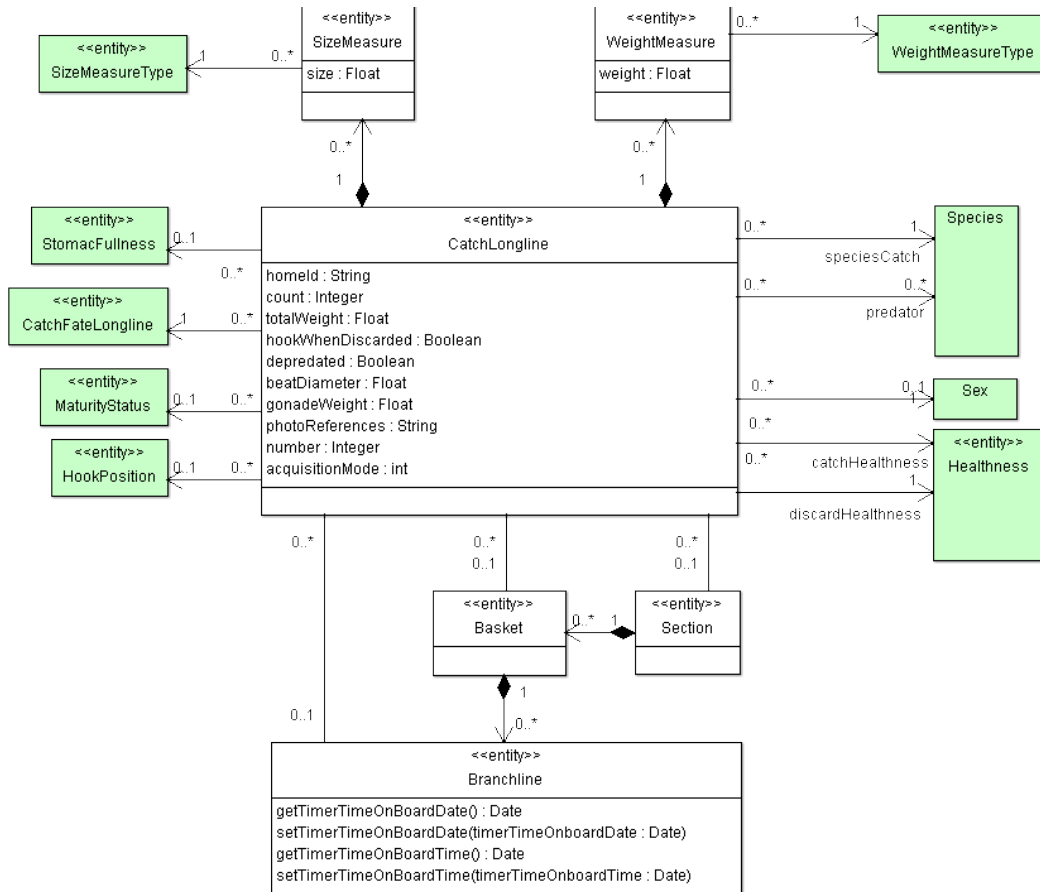


Figure 7. Catch longline view.

### 2.3.6. Time-Depth Recorder (Tdr) view

The *Tdr* view is presented in Figure 8. The data model also manages the location of the TDR deployed on the longline schema on the same principle that it manages catch association with section/basket/branchline. Optionally, a raw data file recorded by the sensors as text or binary file can be stored in the database.

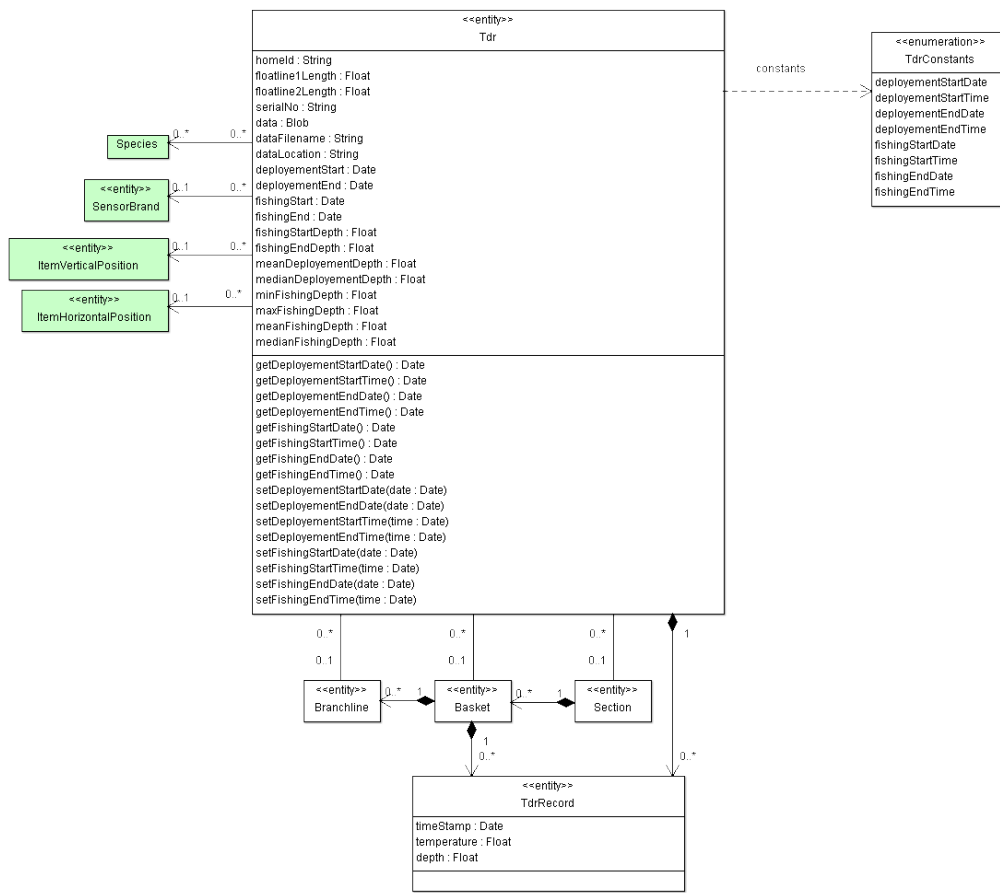


Figure 8. Tdr view.

## 2.4. Data model for purse-seine data

Purse seine and longline observation data structures are quite similar at the trip and activity levels, however, the two models substantially differ in terms of fishing operation and capture. Additionally, the notion of floating object (acting as FAD) only exists in the seine fishery.

### 2.4.1. Set view

Unlike in longline fishing operations, sets in the purse-seine model are geolocated with a unique location. The purse-seine *Set* view is presented in Figure 9. In purse-seine, catch is always recorded by lots of individuals, in tons, not individuals. Length measures are taken for random sample of the catch.

Purse seiner instrumentation allows the observer to collect parameters that are stored in the *Activity* and *Set* entities:

- Physical parameters: water temperature, speed and direction of current and wind,
- Technical parameters about the set: vessel speed, seine net depth when deployed,
- Estimation of biomass parameters by sonar and sounder before the set: weight in tons, its thickness in m, mean and maximum depth.

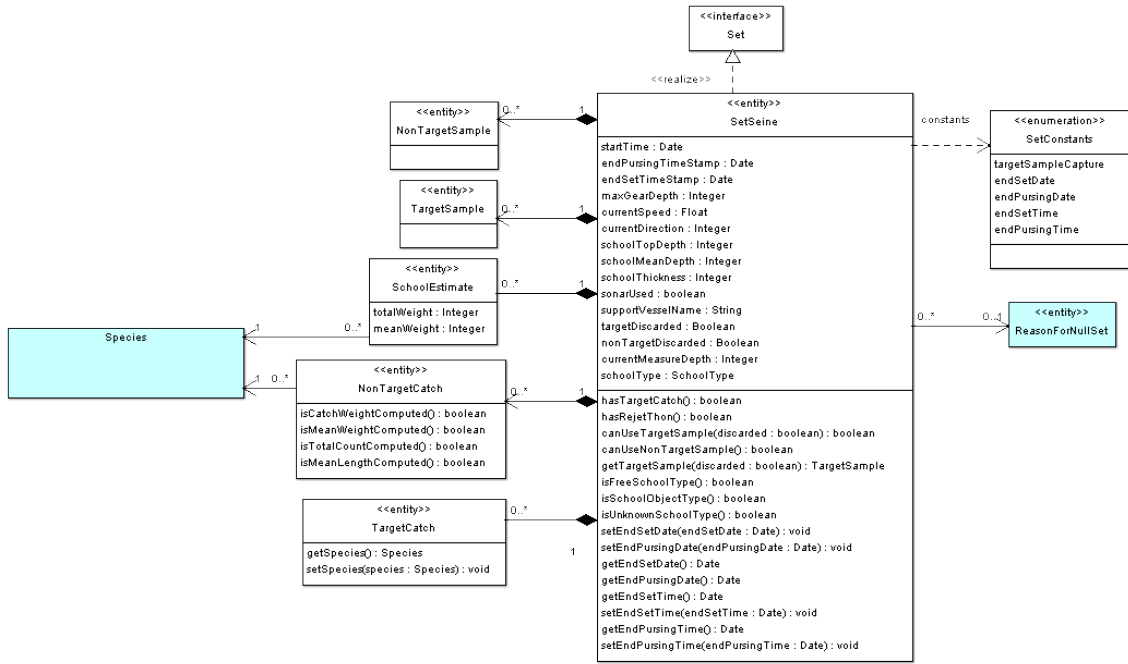


Figure 9. Set view (for purse-seine).

2.4.2. FloatingObject view

The concept of fish-aggregating devices (FAD) is unique to the purse-seine fishery. In the data model, it is taken into account by the *FloatingObject* entity, which is associated to an observation activity and to various parameters: object type, operations performed, estimation of surrounding biomass (Fig. 10).

Moreover, the association between *FloatingObject* and *TransmittingBuoy* allows to store data related to transmitting tags that equip most of artificial FADs.

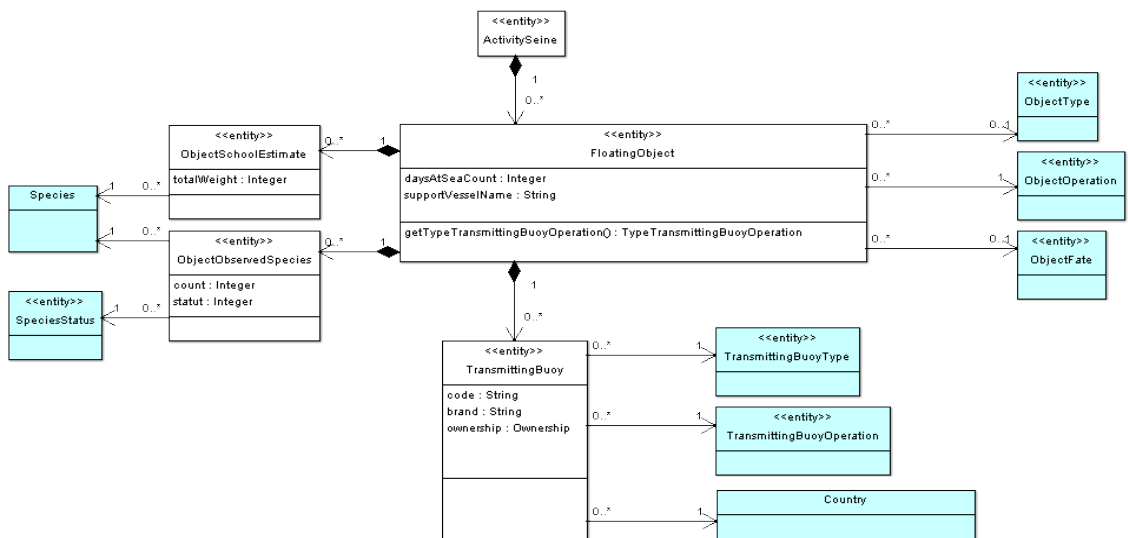


Figure 10. FloatingObject view.

## 2.5. Databases technologies

The data model described above in sections 2.3 and 2.4 are deployed on a central server, as well as on the observers' computers, embedded in the acquisition software. These 2 platforms use different database engines. This decision of using these 2 database engines was guided by criteria of reliability, portability, software transparency and cost-free licenses (open source): the central database runs on *PostgreSQL* while the software-embedded database runs under *H2 Database*. *H2* is written in Java and can run on all platforms for which Java is available; *PostgreSQL* is available for most platforms (*Windows, Linux, Mac, etc.*). Both the central database and the software can run on most platforms available on the market.

### 2.5.1. Acquisition and management software

The development of *ObServe* was guided by three principles: (i) adopting a short workflow, (ii) assisting the observer in its observations and data entry work, and (iii) using sustainable technologies. Figure 11 shows the software interface of *ObServe* used by observers to record capture data.

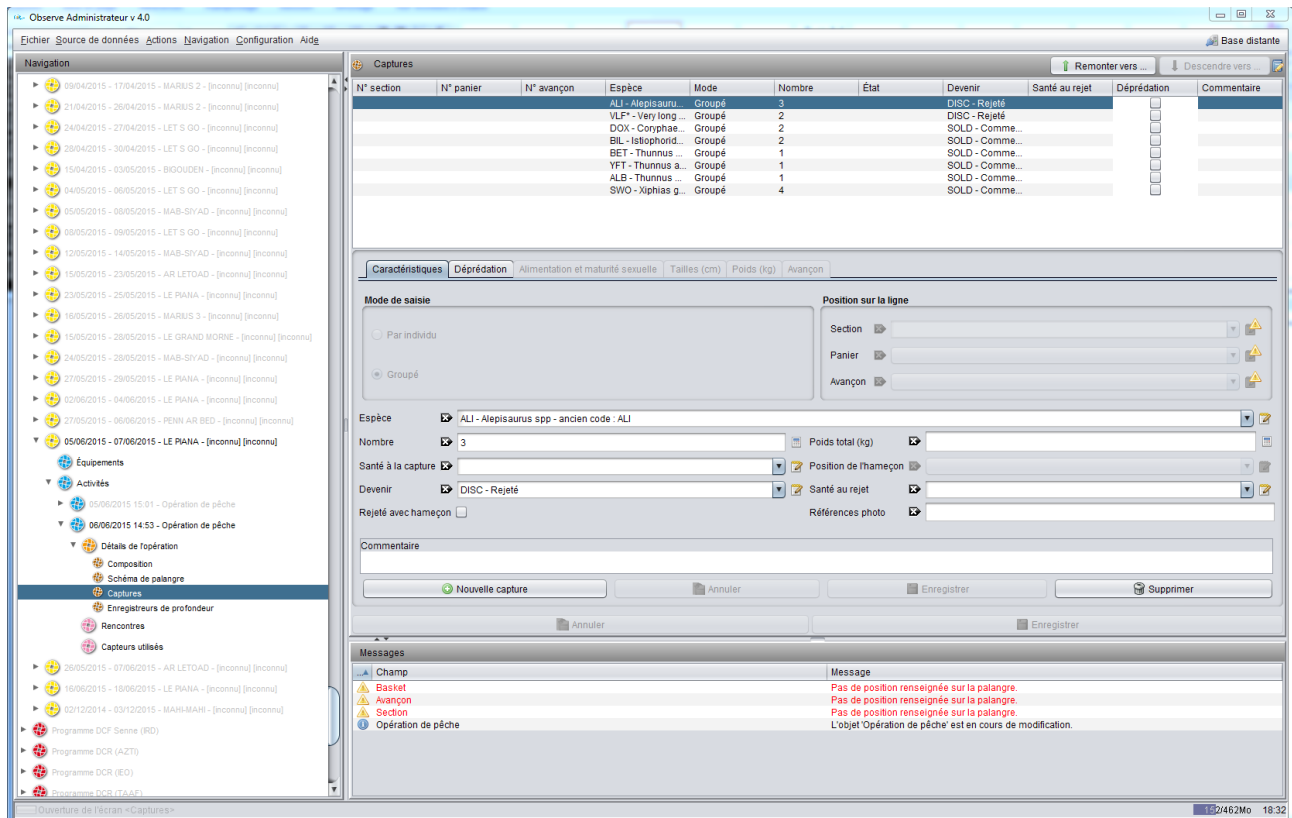


Figure 11. Screenshot of *ObServe* software, catch screen.

### 2.5.2. A short workflow

Workflow is defined as the transformation steps applied to an information to switch from a status to another. This includes both human and computer processing. We believe that the simpler and shorter a workflow, the more it preserves the quality of the information it carries. In *ObServe*, this is achieved through the following principles:

- Software data entry should be done shortly after the observations. The software includes a set of controls applied to input fields resulting in advices or even error messages when the observer fills fields the wrong way. If data are entered shortly after observation the operator can take advantage of his memory to rectify the data. Indeed, as time goes by, corrections will be less and less relevant. For this reason, *ObServe* is designed to be operated at sea, without synchronous network connection to the central database, and to carry out this task when the observer is back on land. Using *ObServe* on board is not mandatory but probably improves the data entry quality,
- The two instances of the data model, on the server and in the software, are identical. For that reason, the second step of the workflow, i.e., importing offline data to the central base, remains simple and easy.

### 2.5.3 Checking and completing data

The software takes advantage of a validation framework (*xworks* library) used to control the values entered by the user. Three levels of control are defined: information, warning and error. Some controls are simple, others implement crosschecking while others are dynamic and make use of reference data (for example, to apply length and weight limits specified for each species).

Controls are primarily used to verify the presence/absence of mandatory information. For example, positions and time stamps of fishing operations are mandatory while the definition of the detailed longline schema is optional. Data entry screens for longline data (like in the data model itself) allow to record all three trip types, including self-reporting trips for which the amount of information is reduced compared to observer-monitored or scientific trips. Mandatory fields are therefore flagged as such according to the type of trip.

When a value is provided, whether it is mandatory or not, it must exactly satisfy limit values or a logic control, otherwise the system will signal an error. For example, vessel speed of a fishing operation is not required, however, if it is filled in, it must be less than or equal to 30 knots. In the same way, length measurements of the different parts of the branchline are not mandatory but must match a defined range when provided.

We believe that these controls significantly improve the quality of the information.

### 2.5.4. Sustainable developments

The development of *ObServe* is built on a Model Driven Architecture (MDA) which principle is to model the application using the Unified Modeling Language (UML) formalism, and use this model to generate a portion of the computer code of the program.

In this project, *ArgoUML* modeling free software (<http://argouml.tigris.org>) was used to create class diagrams. Other libraries were then used to generate code for the Data Access Object (DAO) layer and the

*PostgreSQL* and *H2* database implementations. This approach improves the quality of the object design and facilitates updates since the DAO and the databases will be regenerated when the UML model changes.

The system is fully based on *PostgreSQL* (server side) and *Java* (observer side). These choices provide:

- Independence from computer platforms, as it is important for a system used in the research community, where all types of platforms (*Windows*, *Mac*, *Linux*) can be found, and shared by several organizations that do not necessarily have the same computer equipments,
- Compatibility regarding future versions of operating systems,
- Free licensing of technologies used and possibility to release the project under free GNU GPL license. This aspect is essential as *ObServe* is expected to be shared with many partners.

#### 2.5.5. Ergonomics

Software ergonomics is designed according to two axes. First, the software is ergonomic for data acquisition usage:

- Drop-down lists are powered with auto-completion and can be sorted by codes or labels,
- Help messages are refreshed as the forms are being filled in,
- A map provides a visual overview of the current trip; it can be exported and to be included in reports for example,
- The definition of the detailed longline schema is facilitated by mechanisms of automated generation and contextual menus,
- Safeguards prevent unwanted data alteration,
- Backup and data restoration functions are done in few clicks.

Secondly the system is ergonomic for data managers work:

- The bottom-up synchronization function uploads trips entered offline by the observers into the central database, and manages possible reference data conflicts,
- The top-down synchronization function automates the dissemination of the central repository to the instances of the acquisition software, including references,
- The software allows direct online editing of the trips on the central database.

### 3. Reunion Island longline fishery data in ObServe

#### 3.1. Migration process

Data migration consists in transferring and rewriting data from an old database to a new one. In our case, the original data came from two different databases: *SEALOR*, a *Microsoft Access* database gathering data collected by observers during scientific and commercial longline trips (Bach et al., 2008), and *AutoSampLL*, a *PostgreSQL* database intended to manage self-reporting data collected by professional fishermen (Bach et al., 2013). Both databases differ in their structures, and the destination *ObServe* database also differs from both of them.

As a result, two migration scripts were developed, one for each original database. The 2 migration scripts run on *R* software (R Core-Team, 2015) using *RPostgreSQL* library (Conway et al., 2013). This library includes a driver allowing *R* to access *PostgreSQL* database systems through *SQL* queries. Those migration scripts consisted in two main steps:

- Data extraction: *SEALOR* data were extracted with *mdbtools* (<http://mdbtools.sourceforge.net/>), an open source tool available for Linux platforms used to export *Microsoft Access* data into CSV files, while *AutoSampLL* (already running under *PostgreSQL*) data was read with basic *SQL* SELECT queries,
- Data loading: data were mapped according to *ObServe* destination format and routed into the new system with INSERT queries.

At the end of the migration process, transferred data were verified in order to account for data preservation and accurate data translation, and to check that data supports processes of the new system:

- First, a basic verification was realized by comparing the number of trips, fishing operations and captures between the old systems and the new one.
- Secondly, a comparison was done based on the statistics previously recorded for several longline trips (reports based on the original databases) and the new statistics obtained from *ObServe* for the same trips.

This whole process was repeated several times to improve data quality, optimize migration speed and eliminate redundancies or useless information until the migration procedure was adopted and implemented.

### **3.2. Longline statistics**

#### *3.2.1. Reference data*

A total of 64 longliners were inserted in the database: 63 from Reunion Island, 1 from Mayotte and 1 from Seychelles. In 2015, 24 vessels out of 64 were inactive or not active yet.

Among Reunion Island-based longliners (for which size information was available), 30 vessels were less than 13 m long (LOA: length over all), 11 were about 13 m long, 8 were between 16 and 18 m long, and 11 were between 20 and 25 m long. It should be noted that migrated data were collected from 41 distinct vessels (1 based in Mayotte, 1 in Seychelles, 39 in Reunion and among them, 32 still active). A total of 24 distinct observers that departed from 4 ports (Le Port, Reunion Island; Fort-Dauphin, Madagascar; Mamoudzou, Mayotte; Victoria, Seychelles) were added to the database.

#### *3.2.2. Observation data*

A total of 14 scientific trips (representing 128 fishing sets/68 577 hooks) operated by 7 longliners since 2002, and 72 commercial trips monitored by sea-going observers (representing 580 sets/758 448 hooks) operated by 20 longliners and starting from November 2006, were added to the database (Tab. 1). Scientific trips took place in the Seychelles waters, in the Mozambique Channel and along the east coast of Madagascar. Trips monitored by observers mainly took place in the French and Malagasy EEZs (Fig. 12). Temperature-Depth Recorders (TDR) were deployed during 400 fishing operations with an average rate of 8 TDR loggers per

fishing operation. A total of 28 296 individual captures were recorded during those cruises, and among them, 20 568 individuals were measured.

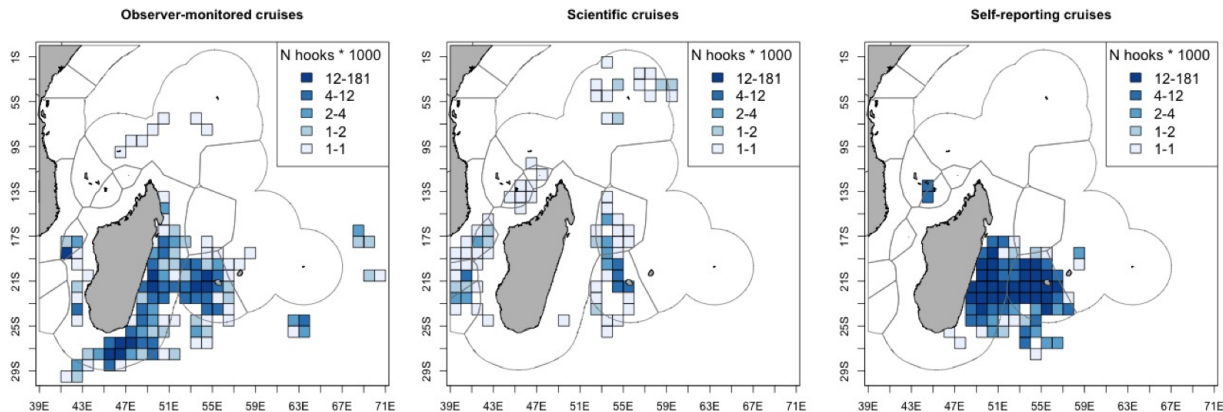


Figure 12. Fishing effort distribution (in number of hooks set) for commercial trips monitored by observers, scientific trips and self-reporting trips.

As for self-reporting data, a total of 266 commercial trips, representing 1 237 fishing sets and 1 554 900 hooks, operated by 30 longliners and starting from May 2011 were inserted in the database (Tab. 1). Self-reported trips essentially took place between Reunion Island and the southeast coast of Madagascar (Fig. 12). TDR devices were deployed at the occasion of 1 353 fishing operations with an average rate of 4 TDR per fishing operation. A total of 44 498 captures were recorded in the database for self-reported data.

Year	Scientific trips			Observer-monitored trips			Self-reported trips		
	N trips	N sets	N hooks	N trips	N sets	N hooks	N trips	N sets	N hooks
2002	1	14	4 353	0	0	0	0	0	0
2003	1	6	2 728	0	0	0	0	0	0
2004	2	21	10 290	0	0	0	0	0	0
2006	0	0	0	1	6	6 600	0	0	0
2007	0	0	0	9	38	40 384	0	0	0
2008	2	13	8 280	4	20	23 726	0	0	0
2009	0	0	0	6	72	95 833	15	15	9 360
2010	3	31	13 575	7	57	71 694	7	7	3 980
2011	4	36	19 159	9	92	114 268	33	127	143 562
2012	1	7	10 192	9	91	120 478	75	398	509 835
2013	0	0	0	12	99	132 231	74	398	512 554
2014	0	0	0	15	105	153 234	62	292	375 609
<b>Total</b>	<b>14</b>	<b>128</b>	<b>68 577</b>	<b>72</b>	<b>580</b>	<b>758 448</b>	<b>266</b>	<b>1 237</b>	<b>1 554 900</b>

Table 1. Yearly number of monitored trips, fishing operations and effort per trip type between 2002 and 2014.



### 3.2.3. Species composition

A total of 72 876 individuals were captured: 29% of tunas (mostly yellowfin *Thunnus albacares*, bigeye *Thunnus obesus* and albacore tuna *Thunnus alalunga*), 19% of swordfish *Xiphias gladius*, 18% of sharks and rays (53% of blue shark *Prionace glauca*), 2% of billfishes, 29% of other species and <1% of protected species (Fig. 13).

Accidental catch of protected species include 70.4% of turtles (47% of loggerhead *Caretta caretta*, 19% of green *Chelonia mydas*, 15% of leatherback *Dermochelys coriacea*, 6% of Ridley *Lepidochelys olivacea*, 6% hawksbill *Eretmochelys imbricata*, 6% of unidentified turtles and 1% of flatback turtles *Natator depressus*), 29.6% were cetaceans (66% of Risso's dolphins *Grampus griseus*, 14% of short-finned pilot whales *Globicephala macrorhynchus*, 11% of unidentified cetaceans, 5% of bottlenose dolphins *Tursiops spp*, 3% of humpback whales *Megaptera novaeangliae*). One unidentified seabird capture was also recorded.

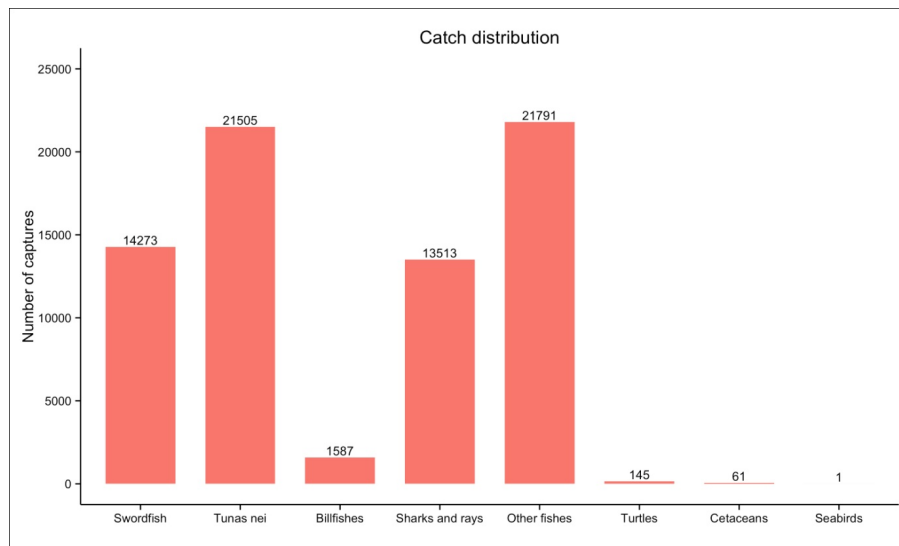


Figure 13. Species composition by species group.

## 3.3. Working tables and automated reports

### 3.3.1. Implementation of views and working tables

A view is the result of a pre-established query on the data stored in the database. Once implemented, the content of a view is automatically updated when new data are inserted in the database. Views present several advantages when analyzing a large and complex database. For instance, views are used to join several tables into a single and simpler one that contains the data useful to the user. Views are custom-made for the user's needs. Views can also provide aggregated results by applying aggregation functions (such as sum or average) on the data.

Based on those considerations, a set of 11 views designed as working tables were implemented:

- 3 reference views that summarize data related to the characteristics of the longliners and species recorded in the database,

- 8 observation views that summarize information related to trips, data location of fishing operations, rigging, catch by lot, individual catch, catch biometry, interactions with depredators or protected species, and TDR deployments.

The second step of the task was to extract information stored in those views (by using a simple *SQL* query: `SELECT * FROM view_name`) to provide 11 working tables that would be saved as *CSV* files on the user's computer. The extraction requires an internet connection to the central database (online) and can be realized with *R* for instance. Once the working tables are saved on the local computer, they can be used offline. For that purpose, we developed a set of *R* scripts that (i) create/update views on the central database, and (ii) extract and save these views as *CSV* files on the user's computer.

### 3.3.2. Automated generation of report outputs

Scientific reports are to be provided on a regular basis in the framework of EU DCF. Those reports are produced on a quarterly basis for the self-reporting program, and for each trip concerning the observer program. These reports aim at summarizing the data collected (spatio-temporal data of the fishing operations, rigging, catch distribution, deployed sensors, interactions with protected species, fishing depth analysis...). Various outputs such as tables and figures are required in those reports.

To produce those outputs, we developed a set of *R* scripts that use the working tables (described in section 3.3.1.) to analyze and generate the outputs of the observer and self-reported data reports. Once produced, the tables and figures can be inserted in the reports.

The observer report script works with trip specified by the user. It produces 11 figures: map of the fishing operations, catch, catch per unit of effort, capture status, discards and depredation distributions, and 5 tables: setting and hauling spatio-temporal locations, rigging, target species CPUE per set, catch per species. One example of the script output is presented in Figure 14.

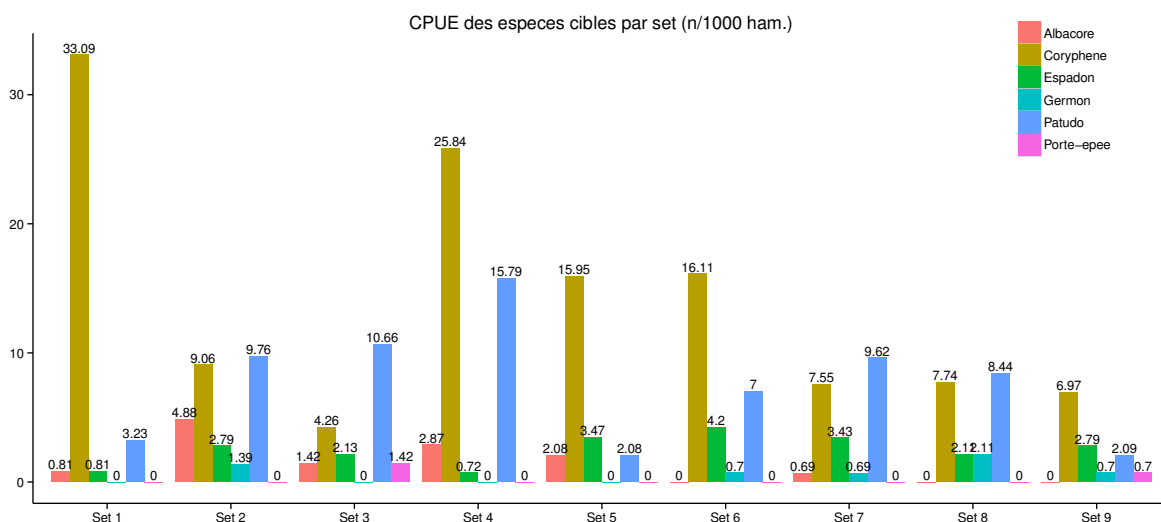


Figure 14. CPUE of target species per set calculated for an observer-monitored commercial trip (SEALOR46).

The self-reporting program report script was intended to work with the time window or the selected trip specified by the user. It generates 3 figures: map of the fishing operations, CPUE distribution per species and EEZ, WPUE distribution per target species, and 8 tables: number of fishing operations and fishing effort by month and by boat, catch distribution, elasmobranchii capture and discard status, protected species capture and discard status, target species CPUE per month, target species CPUE per month and per EEZ, depredation, TDR deployments per boat). One example of the script output is presented in Table 2.

Groupe	Nom scientifique	Nom commun	Code FAO	Total N captures	N conserves	% conserves	N rejetes	% rejetes	N depredates	% depredates	Poids conserve (kg)
Poissons à rostre	<i>Xiphias gladius</i>	Espadon	SWO	320	259	81	61	19	5	2	7265
	<i>Makaira nigricans</i>	Makaire bleu	BUM	23	23	100	0	0	0	0	1110
	<i>Makaira indica</i>	Makaire noir	BLM	16	15	94	1	6	1	6	995
	<i>Tetrapturus angustirostris</i>	Makaire à rostre court	SSP	6	6	100	0	0	0	0	65
	<i>Istiophorus platypterus</i>	Voilier de l'Océan Indien	SFA	5	5	100	0	0	0	0	95
	<i>Tetrapturus audax</i>	Marlin raye	MLS	4	4	100	0	0	0	0	290
Autres poissons	<i>Very long fish nei</i>	Très longs poissons nca	VLF*	232	0	0	232	100	0	0	0
	<i>Alepisaurus spp</i>	Lanciers nca	ALI	65	0	0	65	100	0	0	0
	<i>Lepidocybium flavobrunneum</i>	Escolier noir	LEC	30	0	0	30	100	0	0	0
	<i>Ruvettus pretiosus</i>	Rouvet	OIL	26	0	0	26	100	0	0	0
	<i>Coryphaenidae</i>	Famille Coryphaenidae	DOX	25	21	84	4	16	0	0	137
	<i>Mola spp</i>	Poissons lunes nca	MOP	3	0	0	3	100	0	0	0
	<i>Acanthocybium solandri</i>	Thazard batard	WAH	3	3	100	0	0	0	0	20
Sélaciens	<i>Dasyatis violacea</i>	Pastenague violette	PLS	144	0	0	144	100	0	0	0
	<i>Prionace glauca</i>	Peau bleue	BSH	135	2	1	133	99	0	0	0
	<i>Carcharhinus spp</i>	Requins Carcharhinus nca	CWZ	8	1	12	7	88	0	0	2
	<i>Carcharhinus longimanus</i>	Requin océanique	OCS	4	1	25	3	75	0	0	2
	<i>Sphyrna spp</i>	Requins marteau nca	SPN	3	0	0	3	100	0	0	0
	<i>Galeocerdo cuvier</i>	Requin tigre commun	TIG	1	0	0	1	100	0	0	0
	<i>Mobulidae</i>	Mantes, diables de mer nca	MAN	1	0	0	1	100	0	0	0
	<i>Isurus spp</i>	Taupes	MAK	1	0	0	1	100	0	0	0
	<i>Thunnus alalunga</i>	Gemmon	ALB	167	162	97	5	3	5	3	3204
	<i>Thunnus albacares</i>	Albacore, thon à nageoires jaunes	YFT	92	89	97	3	3	3	3	3105
<i>Thunnus obesus</i>	Patudo, thon obèse	BET	50	46	92	4	8	2	4	990	
Tortues	<i>Chelonia mydas</i>	Tortue verte	TUG	1	0	0	1	100	0	0	0
<b>Total</b>				<b>1366</b>	<b>637</b>		<b>729</b>		<b>16</b>		<b>17280</b>

Table 2. Catch fate distribution per species for self-reported trips (Jan-Mar 2015).

#### 4. Perspectives

In July, 1<sup>st</sup> 2010, IOTC defined an Observer regional Scheme undertaken by CPCs (Resolution 10/04) which major interest is to analyze tuna fisheries data collected in order to adopt an ecosystem approach of the management of tuna fisheries. If this program still needs a better regional coordination, from our point of view the urgency is to provide to CPCs a database tool that allows them to:

- archive collected fisheries data,
- produce reproducible data reporting,
- transmit aggregated data to IOTC in a standard format and thereby offering interoperability between national databases to produce a regional one.

*ObServe* was developed to archive both purse-seine and longline fishery data and produce data reports for given time and space windows. We hope that functions covered by *ObServe* are appealing enough to stimulate its use by other countries.

---

## 5. References

- Bach P., C. Gamblin, V. Lucas, 2008. The effect of bait type on hooking responses of target and non-target species on pelagic longlines: preliminary results from fishing experiments in the Seychelles. *Western Indian Ocean. Journal of Marine. Science*, 7 (2), 151–161
- Bach P., N. Rabearisoa, T. Filippi, S. Hubas, 2008. The first year of SEALOR: Database of SEA going observer surveys monitoring the local pelagic - LOngline fishery based in La Reunion; IOTC-2008-WPEB-13
- Bach P., E. Romanov, T. Filippi, 2009. SWIOFP/ASCLME Project : Mesoscale eddies and large pelagic fish in the Mozambic Channel – Report of monitored longline fishing experiments carried out on board the fishing vessel “Manohal” from 27th of November to 18th of December 2008. IRD/SWIOFP/FFEM Report, 74 p
- Bach P., P.S. Sabarros, 2014. Manuel de l’observateur embarqué à bord des palangriers pélagiques réunionnais. Document technique, Observatoire Thonier IRD, UMR 212 « EME », Programme Européen « Data Collection Framework », 40 p
- Bach P., P.S. Sabarros, L. Le Foulgoc, E. Richard, J.P. Lamoureux and E. Romanov, 2013. Self-reporting data collection project for the pelagic longline fishery based in La Reunion. IOTC–2013–WPEB09–42
- Bach P., N. Rabearisoa, T. Filippi and S. Hubas, 2008. The first year of SEALOR: Database of SEA going observer surveys monitoring the local pelagic - LOngline fishery based in La Reunion. IOTC-2008-WPEB-13
- Conway J., D. Eddelbuettel, T. Nishiyama, S. K. Prayaga, N. Tiffin, 2013. RPostgreSQL: R interface to the PostgreSQL database system. R package version 0.4. <http://CRAN.R-project.org/package=RPostgreSQL>
- R Core Team, 2015. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL <http://www.R-project.org/>
- Sabarros P.S., L. Le Foulgoc, E. Romanov, E. Richard, J.P. Lamoureux and P. Bach, 2013. Catch, bycatch and depredation mapping from self-reporting longline fishery data collected in La Réunion. IOTC–2013–WPEB09–37