This is a preprint of an article accepted for publication in:
Program: Electronic library and Information Systems, Special Issue on Application of Metadata and Semantics in Science, 2015, Emerald (ISSN: 0033-0337), http://www.emeraldinsight.com/loi/prog

Unifying Heterogeneous and Distributed Information about Marine Species through the Top Level Ontology *MarineTLO*

Yannis Tzitzikas[1][2], Carlo Allocca[1], Chryssoula Bekiari[1], Yannis Marketakis[1], Pavlos Fafalios[1][2], Martin Doerr[1], Nikos Minadakis[1], Theodore Patkos[1] and Leonardo Candella[3]

[1] Institute of Computer Science, FORTH-ICS, Greece
[2] Computer Science Department, University of Crete, Greece
[3] Consiglio Nazionale delle Ricerche, CNR-ISTI, Pisa, Italy
{tzitzik, carlo, bekiari, marketak, fafalios, martin, minadakn, patkos}@ics.forth.gr, leonardo.candela@isti.cnr.it

Marine species data are scattered across a series of heterogeneous repositories and information systems. There is no repository that can claim to have all Marine Species data. Moreover, information on marine species is made available through different formats and protocols. Our research aims at providing models and methods that allow integrating such information either for publishing it, browsing it, or querying it. Aiming at providing a valid and reliable knowledge ground for enabling semantic interoperability of marine species data, in this paper we motivate a top level ontology, called **MarineTLO** and discuss its use for creating **MarineTLO**-based warehouses. This approach has been implemented in the context of the iMarine operational European research infrastructure.

1 Introduction

Marine species data are widely distributed with few well-established repositories or standard protocols for their archiving and retrieval. Currently, the various laboratories have in place databases for keeping their raw data, while ontologies are primarily used for metadata that describe these raw data. One of the challenges in the iMarine project[1] is to enable users to experience a coherent source of facts about marine entities, rather than a bag of contributed contents. Considering the current setting where each iMarine source has its own model, queries like *"Given the scientific name of a species, find its predators with the related taxon-rank classification and with the different codes that the organizations use to refer to them"*, cannot be formulated (and consequently nor answered) by any individual source. To formulate such queries, we need an expressive conceptual model, while for answering them we also have to assemble pieces of information stored in different sources. For example, Figure 1 illustrates information about the species Thunnus albacares which is stored in different sources (here FLOD, ECOSCOPE, WoRMS, FishBase and DBpedia, more about these sources in the next section). These pieces of information are complementary, and if assembled properly, advanced browsing, querying and reasoning can be provided.

[1] iMarine, FP7 Research Infrastructures, 2011-2014

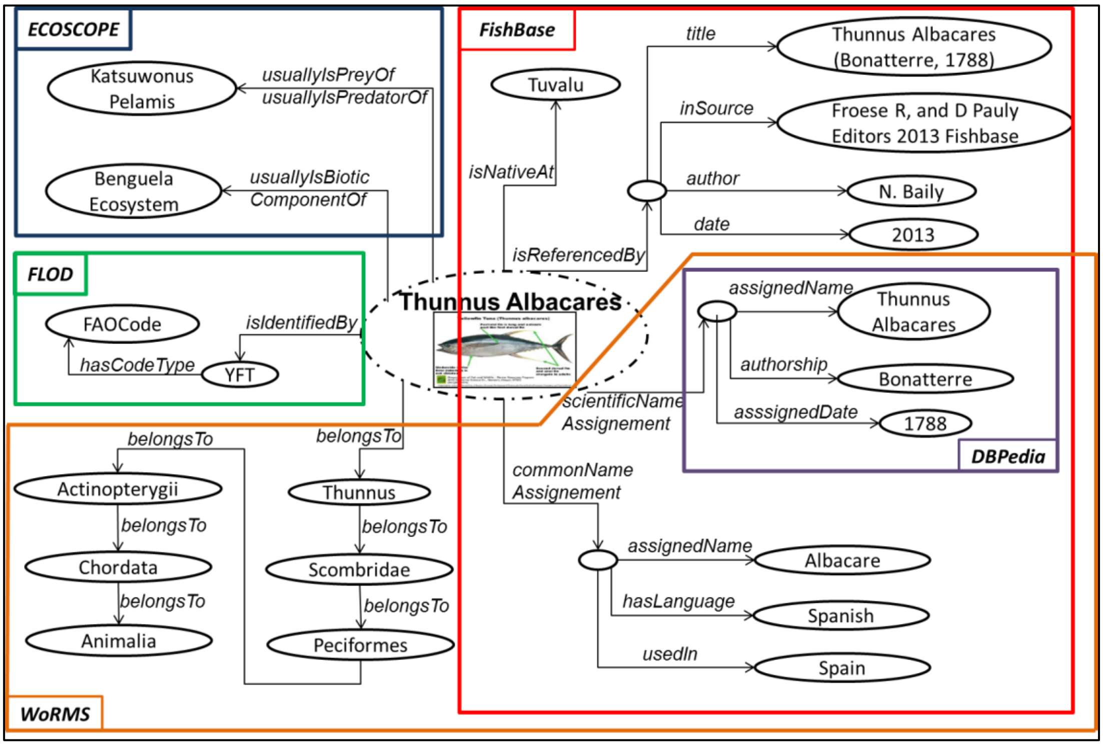


Figure 1: Integrated Information about Thunnus Albacares from five sources

We believe, therefore, that a unified and coherent model for better accessing/reasoning upon and across different marine data sources is a critical and, at the same time, challenging objective, in order to provide a valid and reliable knowledge ground for enabling semantic interoperability of marine data, services, applications and systems. In a nutshell, the key contributions of our work are the following: (a) we identify use cases motivating the need for having harmonized integrated information, (b) we introduce a generic core model, called **MarineTLO**, for schema integration, (c) we describe the mappings between this model and main sources of marine information for building integrated warehouses, (d) we comparatively evaluate two different triplestores for the problem at hand, and (e) we report results regarding the ability of the **MarineTLO**-based warehouse to answer queries which cannot be answered by the underlying sources. To the best of our knowledge, no such warehouse exists.

The rest of this paper is organized as follows. Section 2 discusses the underlying sources and motivating application scenarios, Section 3 describes the proposed approach, Section 4 describes the process for constructing MarineTLO-based warehouses, Section 5 discusses the process for evaluating the ontology, comparatively evaluates two triplestores, and describes the current uses of the MarineTLO-based warehouse. Finally, Section 6 concludes and identifies directions for future work and research.

2 Sources and Motivating Scenarios

In this section, we first describe the main underlying sources (§2.1) and then discuss four motivating scenarios as came up by the organizations participating in iMarine (§2.2).

2.1 Main Underlying Sources

Fisheries Linked Open Data (FLOD) RDF dataset. FLOD, created and maintained by Food and Agriculture Organization (FAO), is dedicated to create a dense network of relationships among the entities of the Fishery domains, and to programmatically serve them to semantic and traditional application environments. The FLOD content is exposed either via a public SPARQL endpoint[2] (suitable for semantic applications) or via a JAVA API to be embedded in consumers' application code. Currently, the FLOD network includes entities and relationships from the domains of Marine Species, Water Areas, Land Areas, Exclusive Economic Zones, and serves software applications in the domain of statistics and GIS.

ECOSCOPE Knowledge Base. IRD[3] offers a public SPARQL endpoint[4] for its knowledge base containing geographical data, pictures and information about marine ecosystems (specifically data about fishes, sharks, related persons, countries and organizations, harbors, vessels, etc.).

WoRMS. TheWorld Register of Marine Species[5] currently contains more than 200 thousand species, around 380 thousand species names including synonyms, and 470 thousands taxa (infraspecies to kingdoms).

FishBase. FishBase[6] is a global database of fish species. It is a relational database containing information about the taxonomy, geographical distribution, biometrics, population, genetic data and many more. Currently, it contains more the 32 thousand species and more than 300 thousand common names in various languages.

DBpedia. DBpedia[7] is a project focusing on the task of converting content from Wikipedia to structured knowledge so that Semantic Web techniques can be employed against it. At the time of writing this article, the English version of the knowledge base of DBpedia describes more than 4.5 million things, containing persons, places, works, species, etc. In our case, we are using a subset of DBpedia's knowledge base containing only fishes (i.e., instances classified under the class http://dbpedia.org/ontology/Fish).

2.2 Motivating Scenarios

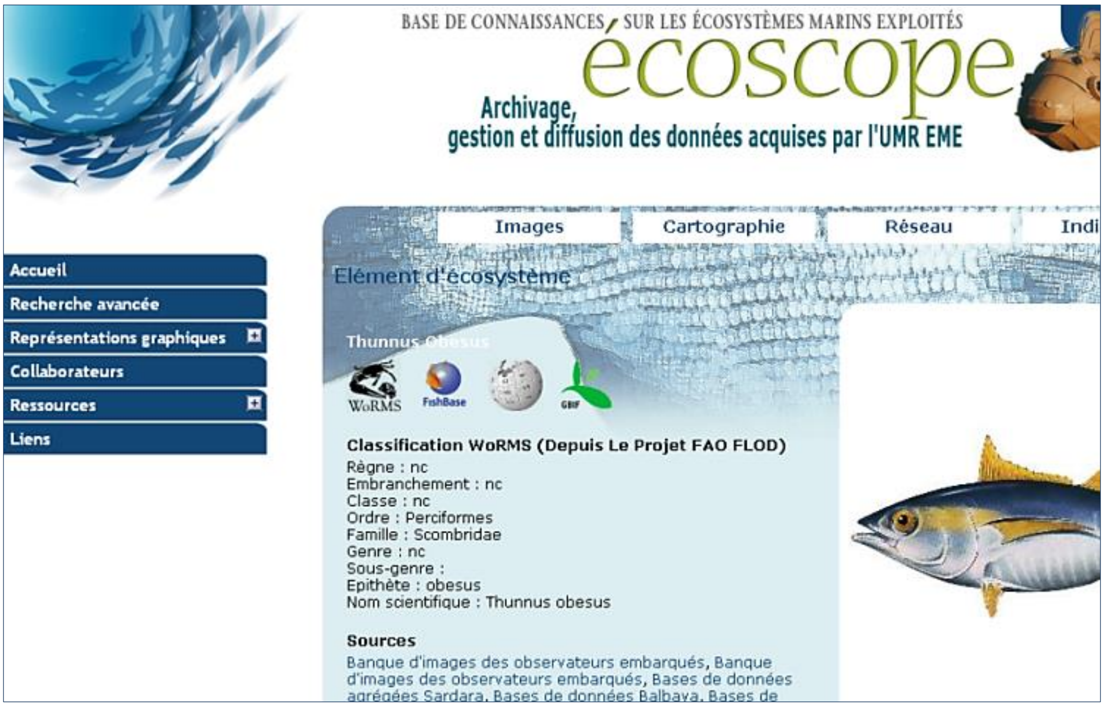
The availability of a top-level ontology for the marine domain would be useful in various scenarios.

For Publishing Linked Data. There is a trend towards publishing Linked Data; consequently a rising issue concerns the structure that is beneficial to use during such publishing. The semantic structure that will be presented can be used by the involved organizations for anticipating future needs for information integration, and thus alleviating the required effort for (post) integration.

Fact Sheets. FactSheetGenerator[8] is an application provided by IRD aiming at providing factual knowledge about the marine domain by mashing-up relevant knowledge distributed across several data sources. Figure 2 shows the results of the current FactSheetGenerator when searching for the species `Thunnus albacares`. Currently, the results are based only on ECOSCOPE and related knowledge stored in other sources (e.g., about commercial codes or taxonomic information) cannot be exploited. The approach that we will present in this

[2] http://www.fao.org/figis/flod/endpoint
[3] Institut de recherche pour le developpement (IRD), France (http://www.ird.fr/)
[4] http://ecoscopebc.mpl.ird.fr/joseki/ecoscope
[5] http://www.marinespecies.org
[6] http://www.fishbase.org
[7] http://dbpedia.org
[8] http://www.ecoscopebc.ird.fr

paper can be exploited for advancing this application, i.e., for providing more complete semantic descriptions.



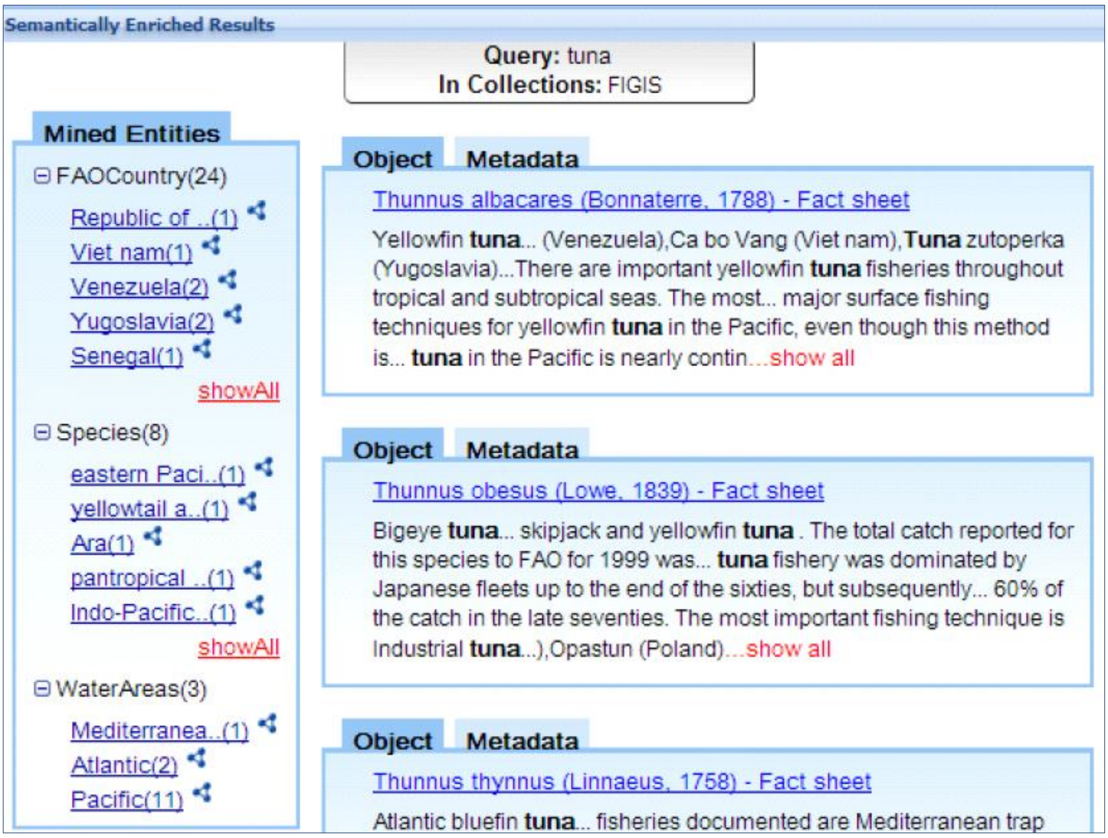
Classification WoRMS (Depuis Le Projet FAO FLOD)
Règne : nc
Embranchement : nc
Classe : nc
Ordre : Perciformes
Famille : Scombridae
Genre : nc
Sous-genre :
Epithète : obesus
Nom scientifique : Thunnus obesus

Sources
Banque d'images des observateurs embarqués, Banque d'images des observateurs embarqués, Bases de données agrégées Sardara, Bases de données Balbaya, Bases de

Figure 2: `Thunnus Albacares` in FactSheetGenerator

For Semantic Post-Processing of the Results of Keyword Search Queries. Another big challenge nowadays is how to integrate structured data with unstructured data (documents and text). The availability of harmonized structured knowledge about the marine domain can be exploited for a *semantic post-processing* of the search results (over dedicated or general purpose search systems). Specifically, the work done in the context of iMarine so far, described in [Fafalios et al., 2012][Fafalios and Tzitzikas, 2013], has proposed a method to enrich the classical (mainly keyword based) searching with *entity mining* that is performed at *query time*. The results of entity mining (entities grouped in categories) complement the query answers with information which can be further exploited by the user in a faceted and session-based interaction scheme [Sacco and Tzitzikas, 2009]. This means that instead of annotating and building indexes for the documents (or web pages), the annotation can be done at query time and using the desired entities of interest. These works show that the application of entity mining over the *snippets* of the top hits of the answers can be performed at real-time, and indicate how semantic repositories can be exploited for specifying the entities of interest and for providing further information about the identified entities.

The initial application within iMarine of this "semantic post-processing" service used FLOD. Figure 3 shows a screen dump of the results for the query `tuna` over a deployment (as a portlet) in an infrastructure where the underlying system is *gcube search* [Simeoni et al., 2007] and the knowledge base is FLOD. The approach presented in this paper has improved this service from various perspectives: more entities can be identified in the results; the system is able to provide more complete information about the identified entities, etc.



Semantically Enriched Results

Query: tuna
In Collections: FIGIS

Mined Entities

⊟ FAOCountry(24)
- Republic of ..(1)
- Viet nam(1)
- Venezuela(2)
- Yugoslavia(2)
- Senegal(1)

showAll

⊟ Species(8)
- eastern Paci..(1)
- yellowtail a..(1)
- Ara(1)
- pantropical ..(1)
- Indo-Pacific..(1)

showAll

⊟ WaterAreas(3)
- Mediterranea..(1)
- Atlantic(2)
- Pacific(11)

Object | **Metadata**

Thunnus albacares (Bonnaterre, 1788) - Fact sheet

Yellowfin **tuna**... (Venezuela),Ca bo Vang (Viet nam),**Tuna** zutoperka (Yugoslavia)...There are important yellowfin **tuna** fisheries throughout tropical and subtropical seas. The most... major surface fishing techniques for yellowfin **tuna** in the Pacific, even though this method is... **tuna** in the Pacific is nearly contin...show all

Object | **Metadata**

Thunnus obesus (Lowe, 1839) - Fact sheet

Bigeye **tuna**... skipjack and yellowfin **tuna** . The total catch reported for this species to FAO for 1999 was... **tuna** fishery was dominated by Japanese fleets up to the end of the sixties, but subsequently... 60% of the catch in the late seventies. The most important fishing technique is Industrial **tuna**...),Opastun (Poland)...show all

Object | **Metadata**

Thunnus thynnus (Linnaeus, 1758) - Fact sheet

Atlantic bluefin **tuna**... fisheries documented are Mediterranean trap

Figure 3: Examples of semantic post-processing of search results within gcube

For Enabling Complex Query Services over Integrated Data. **MarineTLO** can be used as the schema for setting up integrated repositories that offer more complex query services, which cannot be supported by the individual underlying sources. In general, there are two main approaches for building and querying such repositories: the materialized integration approach (or *warehouse* approach), and the virtual integration (or *mediator*) approach (both are described in Section 4). The key point is that in both cases a schema is needed; **MarineTLO** can serve this requirement.

3 MarineTLO-based Integration

3.1 Design Principles

MarineTLO is not supposed to be the single ontology covering the entirety of what exists. It aims at being a *global core model* that i) covers with suitable abstractions the domains under consideration to enable the most fundamental queries, ii) can be extended to any level of detail on demand, and iii) can adequately map and integrate data originating from distinct sources, in a style similar to other related domains [Doerr et al., 2003][Cangemi et al., 2002]. Figure 4 drafts the intended architecture of knowledge models.

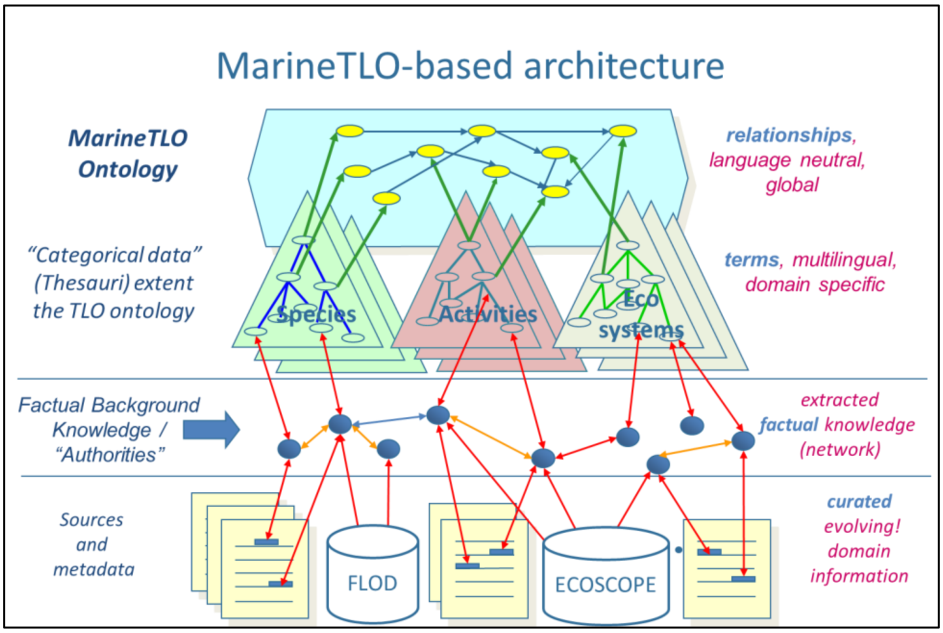

MarineTLO-based architecture
MarineTLO Ontology
"Categorical data" (Thesauri) extent the TLO ontology
Species
Activities
Eco systems
relationships, language neutral, global
terms, multilingual, domain specific
Factual Background Knowledge / "Authorities"
extracted factual knowledge (network)
Sources and metadata
FLOD
ECOSCOPE
curated evolving! domain information

Figure 4: MarineTLO-based architecture

Note that the adoption of a single and coherent core conceptual model has two main benefits: *(a)* reduced effort for improving and evolving it, since the focus is given on one model rather than many [Ibrahim and Pyster, 2004], and *(b)* reduced effort for constructing mappings, since this approach avoids the inevitable combinatorial explosion and complexities that result from pair-wise mappings between individual metadata formats and/or ontologies [Doerr et al., 2003].

Since the marine domain is complex and multiple views or projections should be supported for inference, the **MarineTLO** makes use of (i) categorical and cross-categorical relations as logical derivation of classes and properties of the selected sources, (ii) categories of classes (meta-classes) which support certain type of inference about classes in a way analogous to how classes support certain types of inference about instances and enable the assignment of attribute values to a class. Attention has been given also to the design of **MarineTLO** for preserving *monotonicity*. Since the primary role of **MarineTLO** is the meaningful integration of information in an OpenWorld, it aims to be monotonic in the sense of Domain Theory. That is, the existing constructs and the deductions made from them should remain valid and well-formed, even as new constructs are added to the **MarineTLO**. A particular consequence of this principle is that no class is declared as complement of a sibling concept under a common direct superclass.

Competitive Models. Although many organizations keep marine data, these data are organized based on the needs and activities of the particular organizations. Darwin Core offers a glossary of terms intended to facilitate the sharing of biodiversity information. The philosophy for the development of Darwin Core [Madin et al., 2007][TDWG, 2004][Wilson, 2009][Wieczorek et al., 2012], which intends to keep the standard as simple and open as possible and to develop terms only when there is demand for sharing, is not sufficient. Specifically, the terms are organized into nine categories, often referred to as classes, six of which cover broad aspects of the biodiversity domain (event, location, geological context, occurrence, taxon, and identification). The remaining categories cover relationships to other resources, measurements, and generic information about records. Especially for the record level, Darwin Core recommends the use of a number of terms from Dublin Core (type, modified, language, rights, rights holder, access rights, bibliographic citation, references). Darwin Core was designed to be minimal (only terms shared in common by natural history collections) and flat (no relational structure). A Darwin Core data record leaves the interpretation of the relationships between the whole record and one of its fields to the intuition of the human reader; in other words, it cannot be used to draw logical conclusions (e.g., consistency, equivalence) without human intervention. For instance, if a record level term `dc:ctype` equals to the term “physical object”, then it is understood that the observed record documents a taxon, e.g., a mammal specimen; if on the other hand, the `dc:ctype` is missing, the record is understood to represent the taxon itself. Fields like “prey of” or “predator of” are missing. Also, causally related complex events (or composite events) cannot be described. Darwin Core can serve as a data entry questionnaire. One of the major drawbacks of Darwin Core in the Semantic Web context is the lack of a well-defined ontology, i.e., a formal definition of relationships between the kinds of entities (“core schema”) of the biodiversity domain including its scientific processes. Such an ontology would define the relationships between concepts, such as biological entities, the events that document where and when they occurred, and the processes through which they are identified as being representative of a taxonomic concept. Without rigorous relationships between concepts and the properties that define them, connections between biodiversity data and related semantically rich information, such as literature and genomes, are difficult to traverse and no reasoning can be applied. This creates obstacles to cross-disciplinary semantic inquiry, such as in the Linked Data distributed data community.

Similar approaches have been described also in different domains, i.e., in the medical domain. The Neuroweb Reference ontology [Colombo et al., 2009] is an upper level schema and enables a specific infrastructure to operate over different clinical repositories and to retrieve patients based on a set of specific criteria. This ontology acts as a vocabulary by encoding in a common way the phenotypes (the pathological condition of a patient) of different patients coming from different repositories. A similar approach is followed in the manufacturing domain where a design of top level ontologies is used to provide a ground term for enhancing the collaboration between different labors and partners. In [Mosca et al., 2009] a framework for the development of decision support systems for the engineering domain has been presented. The framework is based on a set of ontologies that describes all the properties of a product so that small and medium enterprises (SME) will be able to easily define the roles of the different labors in the lifecycle of the product (i.e., design, production, testing, etc.).

3.2 The MarineTLO Ontology

For the development and evolution of **MarineTLO** we adopted an iterative and incremental methodology comprising the following steps: (i) ontological analysis of underlying sources, (ii) design, (iii) implementation, and (iv) evaluation. For the implementation we used the OWL Web Ontology Language 2 [Hitzler et al., 2009], while for the needs of evaluation we used the notion of competence queries (described later in this paper). The full version of **MarineTLO**, as well as more information, is available at http://www.ics.forth.gr/isl/MarineTLO.

For the first version of **MarineTLO** we used the descriptions and the data of ECOSCOPE, FLOD and WoRMS sources. As new sources (i.e., DBpedia, FishBase) or new concepts emerged, we updated the **MarineTLO** ontology appropriately. The following list describes its evolution history.

- Version 1 contained 17 classes and 8 unique properties and was designed to capture the scientific names of species and information about predator-prey relationship, coming from ECOSCOPE, FLOD and WoRMS.
- Version 2 contained 57 classes and 22 unique properties. This version captured the same concepts as the previous version, as well as information about WoRMS classification, competitors, images, and species codes coming from FAO and IRD organizations. Furthermore, this version captured specific information about fishes from DBpedia (i.e., scientific and common name of species, images, general description and others).
- Version 3 contained 57 classes and 25 unique properties. This version captured the same concepts as version 2 and furthermore information about water areas, countries, ecosystems, exclusive economic zones, fishing gears, fishing vessels and common names of species. In addition, this version integrated information about the common names of marine species in different language from FishBase.
- Version 4 contained 127 classes and 81 properties. This version captured the concepts of the previous version, as well as information about catch and byCatch[9], biological parameters, statistical indicators (provided by IRD), and publications.

For the needs of the intended applications and the main underlying sources, an extension of the full version is being used. The current version of the extended ontology contains 151 classes and 116 properties. With the name "**`MarineTLO`**", we hereafter refer to this extension. It is organized in two abstraction levels: schema and metaschema. The metaschema aims at providing a method for classifying the schema level in meaningful abstractions, which can be exploited not only for expressing cross-categorical knowledge, but also for aiding the formulation of generic queries. Figure 5 shows the metaclasses (and how they are organized in a `subClassOf` hierarchy), and Figure 6 shows a part of the classes in the class level. Between the classes and the metaclasses there are `instanceOf` relationships (implemented as RDF `typeOf` relationships) which are omitted from the diagram. We use a set of prefixes to declare classes, metaclasses and properties between them. Particularly, we use the prefix BT for declaring the metaclasses (e.g., BT27_Species) and BC for declaring the classes (e.g., BC8_Actor). For the properties we are using the prefix LT for properties between metaclasses (e.g., LT8_usually_belongs_to), LC for properties between classes (e.g., LC13_is_carried_out_by), and LT for cross-categorical links (e.g., LX6_type_was_attributed_by).

[9] http://en.wikipedia.org/wiki/Bycatch

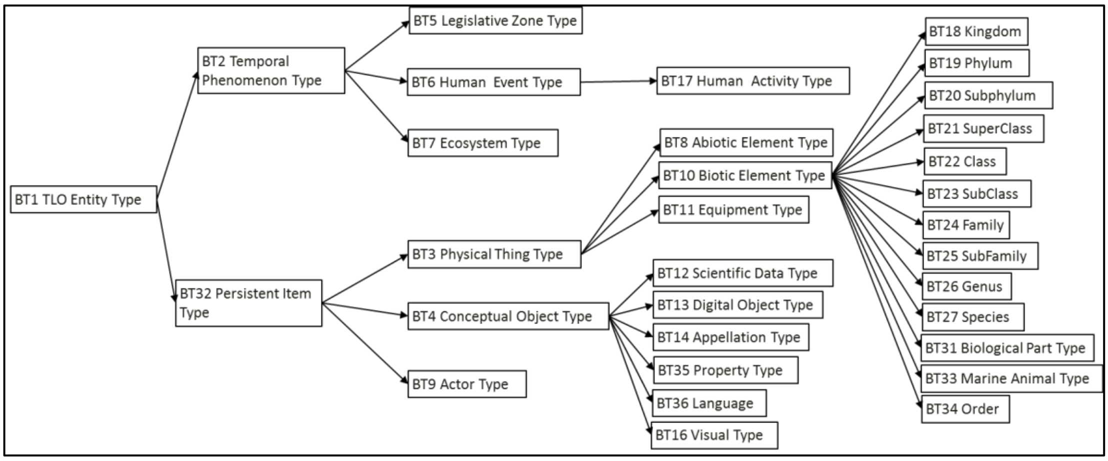


Figure 5: The meta-classes of MarineTLO

The example shown in Figure 1 illustrates how pieces of information that come from different sources and concern one particular species, namely Thunnus Albacares, are assembled. The labels of the frames indicate the used sources. A more detailed example can be seen in Figure 7. The upper part of Figure 7 depicts the scientific name assignment and the lower part shows the taxonomic classification of Thunnus Albacares. Rectangles are used to denote the class name and its corresponding instance (for example, *ns:thunnus_albacares* is an instance of the class **BT27_Species**). In some cases, instead of creating new (or even arbitrary) URIs we are using blank nodes (e.g. the instance of **BT46_Scientific_Name_Assignment**). In those cases, we are using the notation *_:bn* to declare that this particular node is a blank node. Edges are used to denote the properties. Figure 8 shows the same information expressed in RDF. It is evident from this figure that we overcome the issues that arise with new resources; instead of adopting a particular policy for new resources and defining specific namespaces for publishing them, we model them as blank nodes. For example, it is not required to publish a specific URI for the scientific name assignment of Thunnus Albacares, however the information connected to it (i.e., the actual name, the year, the authoritative information) are more than useful.

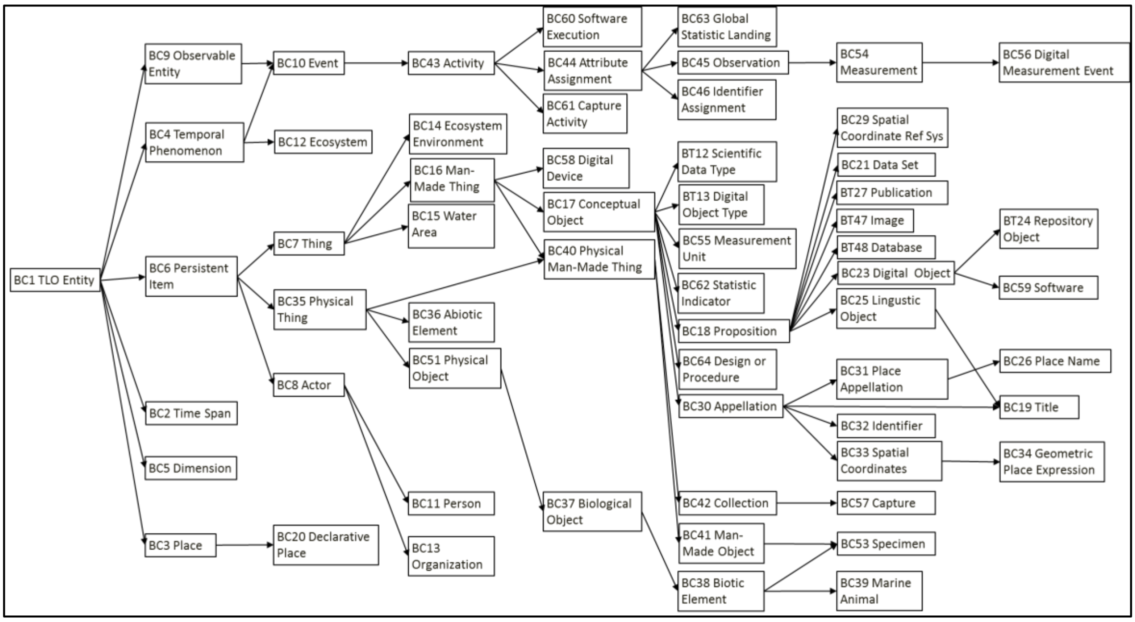


Figure 6: Part of the classes of MarineTLO

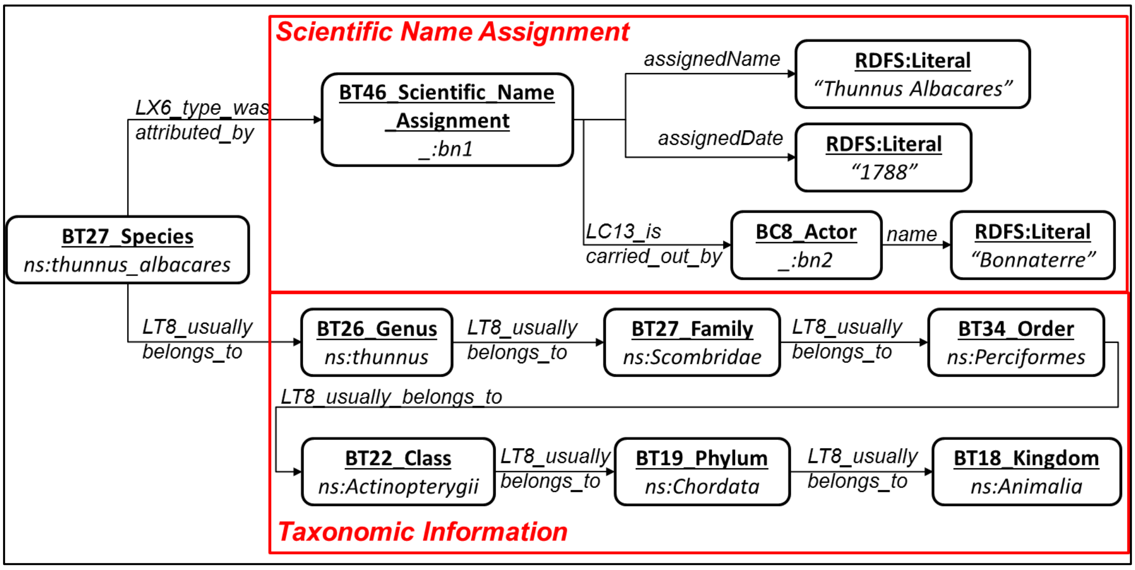


Figure 7: The scientific name assignment and taxonomic information of `Thunnus Albacares`

```
<rdf:RDF xmlns:mtlo="http://www.ics.forth.gr/isl/MarineTLO/v4/marinetlo.owl#"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">

    <mtlo:BT27_Species rdf:about="http://url/thunnus_albacares">
        <mtlo:LX6_assigned_attribute_to_type>
            <mtlo:BC46_1_Scientific_Name_Assignment>
                <mtlo:assignedName> Thunnus Albacares </mtlo:assignedName>
                <mtlo:LC13_is_carried_out_by>
                    <mtlo:BC8_Actor>
                        <mtlo:name> Bonnatere </mtlo:name>
                    </mtlo:BC8_Actor>
                </mtlo:LC13_is_carried_out_by>
                <mtlo:assignedDate> 1788 </mtlo:assignedDate>
            </mtlo:BC46_1_Scientific_Name_Assignment>
        </mtlo:LX6_assigned_attribute_to_type>
        <mtlo:LT8_usually_belongs_to>
            <mtlo:BT26_Genus rdf:about="http://www.ics.forth.gr/isl/thunnus">
                <mtlo:LT8_usually_belongs_to>
                    <mtlo:BT27_Familiy rdf:about="http://www.ics.forth.gr/isl/scombridae">
                        <mtlo:LT8_usually_belongs_to>
                            <mtlo:BT34_Order rdf:about="http://www.ics.forth.gr/isl/perciformes">
                                <mtlo:LT8_usually_belongs_to>
                                    <mtlo:BT22_Class rdf:about="http://www.example/actinopterygii">
                                        <mtlo:LT8_usually_belongs_to>
                                            <mtlo:BT19_Phylum rdf:about="http://url/chordata">
                                                <mtlo:LT8_usually_belongs_to>
                                                    <mtlo:BT18_Kingdom rdf:about="http://url/animalia"/>
                                                </mtlo:LT8_usually_belongs_to>
                                            </mtlo:BT19_Phylum>
                                        </mtlo:LT8_usually_belongs_to>
                                    </mtlo:BT22_Class>
                                </mtlo:LT8_usually_belongs_to>
                            </mtlo:BT34_Order>
                        </mtlo:LT8_usually_belongs_to>
                    </mtlo:BT27_Familiy>
                </mtlo:LT8_usually_belongs_to>
            </mtlo:BT26_Genus>
        </mtlo:LT8_usually_belongs_to>
    </mtlo:BT27_Species>
</rdf:RDF>
```

Figure 8. The scientific name assignment and taxonomy of `Thunnus Albacares` in RDF

4 On Constructing MarineTLO-based Warehouses

4.1 Integration Approaches

In general, there are two main integration approaches for such repositories: the materialized integration approach (or warehouse approach), and the virtual integration (or mediator) approach.

Materialized Approach The materialized approach relies on a central repository (RDF triplestore in our case) where all data are to be stored. *Mappings* (in the broad sense) are exploited to *extract* information from data sources, to *transform* it to the target model and then to *store* it at the central repository. Over such a repository more complex queries can be answered.

It is good practice not to modify extracted information after each transformation except for the use of common identifiers. Rather, any need for updating individual information is covered by requesting source providers to make updated sources available. There are some important issues that should be taken into account for designing and maintaining a data warehouse. Firstly (design phase), the information from each source that is going to be used should be selected. Specific views over the sources should be chosen in order to be materialized. Next (maintenance phase), issues should be tackled concerning the warehouse initial population by the source data and the update of the data when sources are refreshed. The notion of *graph spaces* of RDF triplestores can alleviate this problem. The great advantage of materialized integration is its flexibility in transformation logic, the decoupling of the release management of the integrated resource from the management cycles of the sources, and the decoupling of access load from the source servers. The method that we will present can be used for setting up such repositories.

Moreover, the availability of a materialized repository is beneficial for applying entity matching techniques (e.g., see [Noessner et al., 2010]) since more information about the domain entities is available, while the application of these techniques is significantly faster than applying them without having a repository (i.e., by fetching information from the network).

Virtual Approach. On the other hand, the virtual integration approach does not rely on a central repository but leaves the data in the original sources. Mappings (in the broad sense) are exploited to enable *query translation* from one model to another. Then, data from disparate sources is *combined* and returned to the user. The mediator (a.k.a. integrator) performs the following actions. First, it receives a query formulated in terms of the unified model/schema and decomposes the query into *sub-queries*. These queries are addressed to specific data sources. This decomposition is based on the mappings generated between the unified model and the source models, which play an important role in sub-queries' execution plan optimization. Finally, the sub-queries are sent to the wrappers of the individual sources, which transform them into queries over the sources. The results of these sub-queries are sent back to the mediator. At this point, the answers are merged and returned to the user. Besides the possibility of asking queries, the mediator has no control over the individual sources. The great advantage (but in some cases disadvantage) of virtual integration is the real-time reflection of source updates in integrated access. As regards system's complexity (complexity of query rewriting and of execution planning), this depends on the structural complexity of the global view and the differences between this view and that of the underlying models. The higher complexity of the system (and the quality of service demands on the sources) is only justified if immediate access to updates is indeed required.

4.2 The Marine-TLO based Warehouse

We have been investigating the materialized (warehouse) approach. Specifically, we coded the **MarineTLO** ontology using OWL 2 and set up a repository using two different triplestores which are described in the sequel. Apart from **MarineTLO**, the repository contains the entire FLOD fetched from its SPARQL endpoint, the entire ECOSCOPE downloaded by its web site, and parts of WoRMS extracted using SDDS and TLO Wrapper, part of FishBase extracted using FishBaseReaper and part of DBpedia fetched from its public endpoint.[10] Figure 9 displays the current **MarineTLO**-based warehouse.

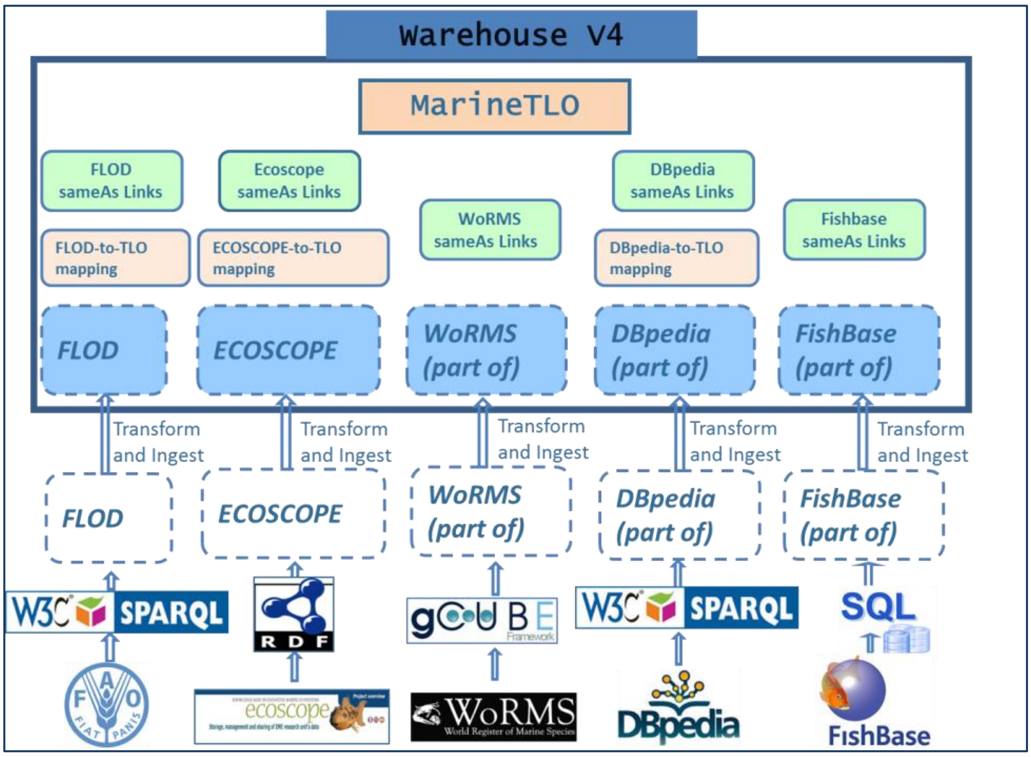


Figure 9: The current version of **MarineTLO**-based warehouse

Used Triplestores. We have comparatively evaluated two different triplestores: OWLIM-Lite[11] and OpenLink Virtuoso[12]. The first has been designed for medium data volumes (less than 100 million statements). It contains a persistence layer, however reasoning and query

[10] SDDS, TLOWrapper and FishBaseReaper will be described in the subsequent sections.
[11] http://owlim.ontotext.com/
[12] http://virtuoso.openlinksw.com/

evaluation are being performed entirely in main memory. On the other hand, Virtuoso supports backward chaining reasoning, meaning that it does not materialize all inferred facts, but computes them at query level. Practically, this means that transitive relations (i.e., `rdfs:subClassof`, `rdfs:subPropertyOf`, `owl:equivalentClass` etc.) are not physically stored in the knowledge base, but they are added to the result set during query answering. In §5.2 we comparatively evaluate these triplestores.

4.3 MarineTLO-based Mappings

For extracting information from the underlying sources and associating them with **MarineTLO**-based descriptions, we use *mappings*. In general, what we call mapping comprises: extensions to the metaschema, extensions to the schema, `rdfs:subClassOf`, `rdf:subPropertyOf`, `owl:equivalentClass` relationships between the elements of MarineTLO and the schema at hand, plus some inference rules. Below, we sketch the defined mappings. For instance, the ECOSCOPE-2-**MarineTLO** mapping consists of `rdfs:subClassOf` and `rdfs:subPropertyOf` like those shown in Figure 10. The DBpedia-2-**MarineTLO** mapping contains analogous relationships. Note that we do not use any mappings for FishBase and WoRMS, since the software tools we are using for transforming data from these sources (see §4.4), directly produce instances which are expressed with respect to **MarineTLO** descriptions.

```
(tlo:EcoscopeSpecies, rdfs:subClassOf, tlo:TLOSpecies)
(eco:fish, rdfs:subClassOf, tlo:EcoscopeSpecies)
(eco:is_predator_of, rdfs:subPropertyOf, tlo:usuallyIsPredatorOf)
(eco:is_prey_of, rdfs:subPropertyOf, tlo:usuallyIsPreyOf)
(eco:biotic_component_of, rdfs:subPropertyOf, tlo:usuallyIsComponentOf)
(eco:used_data_source, rdfs:subPropertyOf, tlo:isReferencedBy)
```

Figure 10: Mappings between Ecoscope and MarineTLO

However, in FLOD any resource is an instance of `CodedEntity`, and for distinguishing a vessel (e.g. `vessel_289`) from a species (e.g. `thunnus albacares`) we need to go one step further and look at its code. For instance, we can distinguish FAOSpecies as follows:

$$FAOSpecies=\{x \mid CodedEntity(x) \text{ and } (\exists y\ isClassifiedByCode(x,y) \text{ and } SpeciesCode(y))\}$$

The required mapping can be defined using `owl:Restriction`. This is supported by OWLIM, but it is not supported by Virtuoso. For the latter, we can express this mapping through the following SPARQL INSERT query:

```
INSERT {
 ?x rdf:type < http://www.ics.forth.gr/isl/MarineTLO/v4/marinetlo.owl#FLOD_Species> }
WHERE {
 ?x rdf:type <http://www.fao.org/figis/flod/onto/codedentity.owl#CodedEntity> .
 ?x <http://www.fao.org/figis/flod/onto/codedentityclassification.owl#isClassfiedByCode> ?y .
 ?y rdf:type <http://www.fao.org/figis/flod/onto/linneanspecies.owl#SpeciesCode> }
```

Figure 11: Expressing OWL Restriction as a SPARQL Insert query

4.4 Software for Transforming Instances from Heterogeneous Sources to MarineTLO

In some cases the data of the underlying sources (i.e., FishBase, WoRMS) is described in different formats. In these cases we have to transform the data to RDF. For this reason, we have implemented particular tools for carrying out these transformations, which are described below.

FishBaseReaper. FishBase contains information in relational databases. We have implemented a tool, called FishBaseReaper, that extracts data from these databases and transforms them to RDF instances according to MarineTLO. The tool takes as input a list of concepts of interest (scientific name, ecosystems, bibliographic information), connects to the relational databases of FishBase, and produces as output files (in N-Triples format) that contain the extracted information with respect to **MarineTLO** classes and properties. There is also the option of using a specific URI prefix for all the extracted entities, or different URIs prefixed according to the type of each entity (i.e., use of the namespace http://www.ics.forth.gr/isl/MarineTLO/Species for marine species and http://www.ics.fort.gr/isl/MarineTLO/PlaceName for countries).

The Species Data Discovery Service (SDDS). The Species Data Discovery Service [Candela et al., 2014-1], SDDS for short, is a gCube service [Candela et al., 2008] specifically conceived to provide its users with a single access point to species data, both occurrence data and nomenclature data, hosted by a number of distributed databases. It is a plugin-based mediator service for key species data databases including GBIF and OBIS for occurrence data, Catalogue of Life, OBIS, Interim Register of Marine and non-marine Genera (IRMNG), ITIS, NCBI, and WoRMS for nomenclature data.

Given that the set of databases that the service interfaces with is open, it is sufficient to implement a dedicated plugin to (i) transforming the query for species data expressed in a domain specific query language into an equivalent query supported by the specific data provider, (ii) submit the transformed query to the data provider, and (iii) transform the results into the common data format envisaged by the SDDS. Details on the domain specific query language and the unifying data format characterizing SDDS are discussed in [Candela et al., 2014-2]. Here it is worth to highlight that (i) the query language is essentially based on species names (scientific and common names) and supports directives for automatic query expansion based on known species names, (ii) the resulting records are annotated with details on data provenance produced accordingly to the policies of each data source, and (iii) the resulting records can be produced according to known formats and standards including DarwinCore [Wieszorec et al., 2012]. SDDS is offered both via a web-based user interface and a web-based API for programmatic access.

MarineTLO Wrapper. We implemented a tool that uses SDDS API and transforms the fetched information into descriptions structured according to the **MarineTLO**. Its functionality is performed in two phases: during the first phase, it takes as input a list of scientific names to be retrieved and the data sources to be searched and submits the query to SDDS. The output is a Darwin Core Archive (DwC-A) file, containing the classifications of the given input. During the second phase the tool parses the DwC-A archives and produces the descriptions according to **MarineTLO**. Finally, the data are exported in RDF or NTRIPLES format.

4.5 Transformation and Entity Matching Rules

Transformation Rules. Some data can be stored into the warehouse as they are fetched, while others may need to be transformed. For example, a literal may need to be transformed into a URI, or to be split for using its constituents, or an intermediate node may need to be created (e.g. instead of (x,hasName,y) to have (x,hasNameAssignement,z),(z,name,y),(z,date,d). In order to handle such cases, we created a number of transformation rules that are applied to the fetched data, before its ingestion to the warehouse. The following table shows some of the transformation rules we applied for the warehouse; for each transformation rule, the upper row shows the initial expression and the lower one shows the transformed expression.

Table 1: Transformation Rules

1	The string “Bonnattere, 1788” for the entry Thunnus Albacares in Worms
	<ns:thunnus_albacares> <mtlo:LX6_type_was_attributed_by> _:bn1 _:bn1 <mtlo:is_carried_out_by> _:bn2 _:bn2 <mtlo:name> “Bonnaterre” _:bn1 <mtlo:assignedDate> “1788”
2	The value (?val) of the property skos:prefLabel for every instance (?inst) of the class http://www.ecoscope.org/ontologies/ecosystems_def/fish
	?inst <mtlo:type_was attributed_by> _:bn1 _:bn1 <mtlo:assignedName> ?val
3	?x <mtlo:usually_is_predator_of> ?z ?y <mtlo:usually_is_predator_of> ?z ?x != ?y
	?x <mtlo:usually_is_competitor_of> ?z

SILK Rules. We created same-as relationships between the entities using an entity matching tool called SILK link[13]. Specifically, i) we inspected the connectivity between the sources, ii) we formulated a number of silk same-as rules, iii) we applied these rules to the sources and iv) we imported the produced same-as relationships into the warehouse. The reason for applying these rules is that they increase the connectivity of the resulting warehouse (this aspect will be discussed in detail later). Table 2 shows some indicative SILK rules[14].

Table 2: Rules for creating owl:sameAs links using SILK

#	Value from Source A	Value from Source B	Example

1	`wormsId` attribute from ECOSCOPE	Integer part of `hasTaxonID` attribute from WoRMS	`wormsId: 127027`
			`hasTaxonId: WoRMS:127027`

2	`prefLabel` attribute (in lower case) from ECOSCOPE	`label` attribute (in latin) from FLOD	`prefLabel: Thunnus albacares`
			`label: “thunnus albacares”@la`

3	`altLabel` attribute from ECOSCOPE	`label` attribute from FLOD	`altLabel: “yellowfin tuna”@en`
			`label: “yellowfin tuna”@en`

4	`prefLabel` attribute from ECOSCOPE	`binomial` attribute from DBpedia	`prefLabel: Thunnus albacares`
			`binomial: Thunnus albacares`

5	`label` attribute tokenized using “ from FLOD	`binomial` attribute (to lower case) from DBpedia	`label: “thunnus albacares”@la`
			`binomial: Thunnus albacares`

6	`label` attribute tokenized using “ from FLOD	WoRMS species URI after removing the namespace, the taxon id, replacing underscores ‘_’ with spaces and converting it to lower case	`label: “thunnus albacares”@la`
			`URI: http://www.marinespecies.org/ entities/ WoRMS:127027/ Thunnus_Albacares`

7	`binomial` attribute from DBpedia	WoRMS species URI after removing the namespace, the taxon id, replacing underscores	`binomial: Thunnus albacares`
			`URI: http://www.marinespecies.org/ entities/ WoRMS:127027/`

[13] http://wifo5-03.informatik.uni-mannheim.de/bizer/silk/

[14] The full list of the SILK rules that are being used for constructing the **`MarineTLO`**-based warehouse can be found at **`MarineTLO`** website http://www.ics.forth.gr/isl/MarineTLO/.

		'_' with spaces and converting it to lower case	`Thunnus_Albacares`

8	`binomial` attribute from DBpedia	FishBase species URI after removing the namespace and replacing underscores '_' with spaces	`Binomial: Thunnus albacares`
			`URI: http://www.fishbase.org/entity#thunnus_albacares`

9	WoRMS species URI after removing the namespace, the taxon id, replacing underscores '_' with spaces and converting it to lower case	FishBase species URI after removing the namespace and replacing underscores '_' with spaces	`URI: http://www.marinespecies.org/entities/ WoRMS:127027/Thunnus_Albacares`
			`URI: http://www.fishbase.org/entity#thunnus_albacares`

10	`label` attribute tokenized using " from FLOD	FishBase species URI after removing the namespace and replacing underscores '_' with spaces	`label: "thunnus albacares"@la`
			`URI: http://www.fishbase.org/entity#thunnus_albacares`

11	`prefLabel` attribute from ECOSCOPE	FishBase species URI after removing the namespace and replacing underscores '_' with spaces	`prefLabel: Thunnus albacares`
			`URI: http://www.fishbase.org/entity#thunnus_albacares`

4.6 Assessing the Connectivity of Information

From this activity, we observed that the data fetched from the sources are in many cases problematic (consistency problems, duplicates, wrong values). We noticed that placing them together in a warehouse makes easier the identification of such errors. Furthermore, the availability of the warehouse enables defining `sameAs` connections by exploiting transitively induced equivalences, and can be produced by exploiting SILK matching rules, like the ones described in Section 4.5. In any case, the inspection of the repository for detecting the missing connections that are required for satisfying the needs of the competence queries is an important requirement. To this end, we have devised some metrics for quantifying the value of the warehouse and the value (contribution) of each source to the warehouse. These metrics are described in detail in [Tzitzikas et al., 2014-1], while a vocabulary that allows the representation, exchange and querying of such measurements is described in [Mountantonakis et al., 2014].

4.7 Handling the Provenance

After ingesting data coming from several sources in the warehouse we can still identify their provenance. We support four levels of provenance: (a) at *conceptual modeling level*, (b) at *URIs and values level*, (c) at *triple level*, and (d) at *query level*.

As regards (a), **`MarineTLO`** models the provenance of species names, codes etc. (who and when assigned them). Therefore, there is no need for adopting any other model for capturing provenance (e.g. OPM [Moreau et al., 2011]). As regards (b), we adopt the namespace mechanism for reflecting the source of origin of an individual. For example, the URI `http://www.fishbase.org/entity#thunnus_albacares` denotes that this URI has been derived from FishBase. Furthermore, during the construction of the warehouse there is the option of applying a uniform notation @source to literals (where source can be FLOD, ECOSCOPE, WoRMS, FishBase, DBpedia). As regards (c), we store the triples from each source in a separate graph space. This is useful not only for provenance reasons, but also for refreshing parts of the warehouse, as well as for computing the connectivity metrics that have been described previously. Finally, as regards (d), we have implemented a framework, called

MatWare that is described below, which offers a query rewriting functionality that exploits the graph spaces, and returns the sources that contributed to the query results. The provenance at *URIs and values level* is just an alternative way of modelling provenance. We used this approach for modeling the scientific names of the species in the first versions of the warehouse. In subsequent versions we used the triple level provenance which allows storing data coming from different sources using different graph spaces. In this case the provenance of all the contents of a source is being derived from the graph space. An extensive discussion about provenance in **`MarineTLO`**-based warehouses can be found at [Tzitzikas et al., 2014-2].

4.8 Connecting the Pieces

We developed a framework, called *MatWare* [Tzitzikas et al., 2014-2], that automates the construction, maintenance and quantitative evaluation of warehouses based on **`MarineTLO`**. Figure 12 illustrates the warehouse construction and evolution process, as supported by *MatWare*.

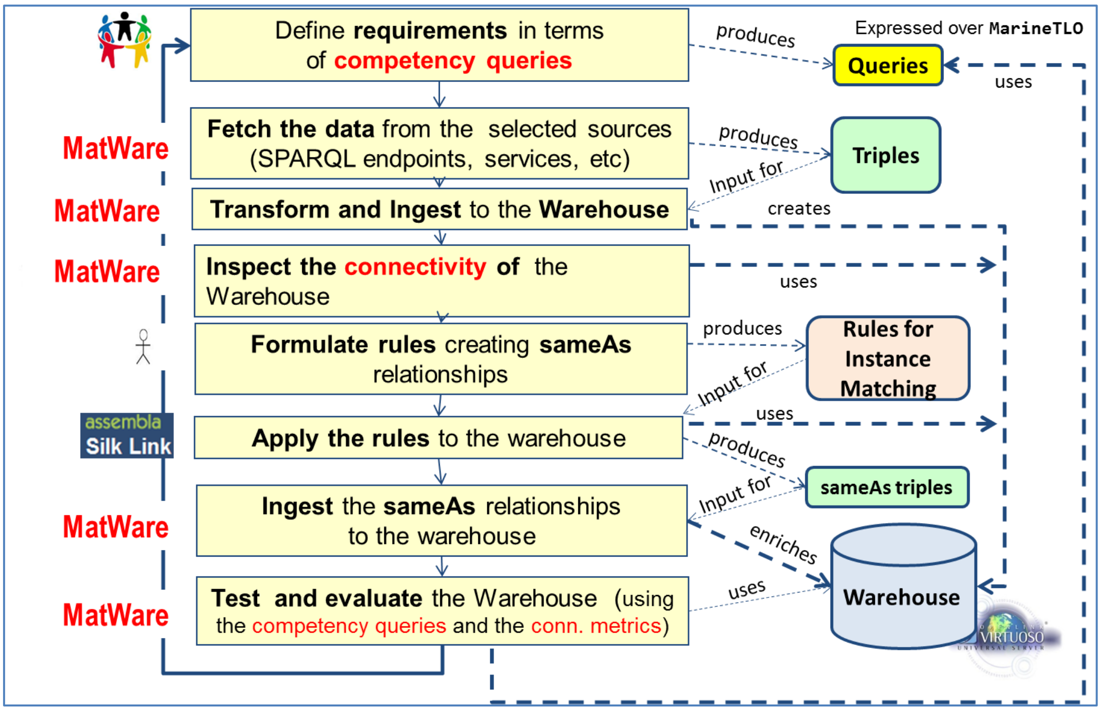

Define requirements in terms of competency queries
Fetch the data from the selected sources (SPARQL endpoints, services, etc)
Transform and Ingest to the Warehouse
Inspect the connectivity of the Warehouse
Formulate rules creating sameAs relationships
Apply the rules to the warehouse
Ingest the sameAs relationships to the warehouse
Test and evaluate the Warehouse (using the competency queries and the conn. metrics)
MatWare
MatWare
MatWare
assembla Silk Link
MatWare
MatWare
produces
Expressed over MarineTLO
Queries
uses
produces
Triples
Input for
creates
uses
produces
Rules for Instance Matching
Input for
uses
produces
sameAs triples
Input for
enriches
uses
Warehouse
VIRTUOSO

Figure 12: The warehouse construction and evolution process supported by *MatWare*.

5 Evaluation and Current Uses of the `MarineTLO`-based Warehouse

5.1 Evaluating the `MarineTLO`-based Warehouse through Competence Queries

For evaluating the structuring of **`MarineTLO`** and the process used for creating the **`MarineTLO`**-based repository, we had to investigate whether they offer the required abstractions for (a) adequately modeling the domain, (b) hosting information coming from different sources, and (c) allowing answering useful queries which cannot be answered by the individual underlying sources. For the latter, we formed a collection of competence queries in collaboration with the involved partners and their priorities. Table 3 shows some fundamental concepts that exist in the competence queries. The columns at the right show which of them are answerable by the underlying sources. We should note that the real *competence queries* include queries that combine more than one of the listed concepts, like the complex query that was described in the introduction of this paper, e.g., "*I want the taxonomic information of the predators of a particular species with the different codes that the organizations use to refer to them*". This particular query requires sources that contain information about: (a) scientific names, (b) species taxonomy, (c) predators, and (d) codes (usually provided by FLOD/FAO). Such queries cannot be answered by any particular source (which is also evident for the particular example from the contents of Table 3), but can now be answered by the **MarineTLO**-based Warehouse that contains the required sources. This is the concrete evidence of the benefits offered by the integrated model. A table showing the competence queries we used and their corresponding SPARQL expression can be found at **MarineTLO** website.

Table 3: Basic Queries

Concepts	ECOSCOPE	FLOD	WoRMS	DBpedia	FishBase
Species Taxonomy			✓	✓	✓
Scientific/Common Names	✓	✓	✓	✓	✓
Authorships			✓	✓	✓
Predators	✓				
Ecosystems	✓				
Countries					✓
Water Areas		✓			✓
Vessels	✓	✓			
Gears	✓	✓			
EEZ[15]		✓			
Bilbiography	✓		✓		✓
Statistical Indicators					✓

5.2 Comparison of Different Triplestores

Table 4 shows the sizes in triples of the contents of the OWLIM and Virtuoso repositories for the first version of the **MarineTLO** and its corresponding sources. The first contains in total 10.8 million triples. This number includes the inferred triples, since this repository materialized them. The creation of the repository from scratch (by loading the corresponding files) takes around 30 minutes. The time is short because the used edition of OWLIM loads everything in main memory. In Virtuoso the number of triples is significantly lower, because the inferred triples are not stored. The creation in this case takes 4h and 20 minutes[16]. The execution of the INSERT query (needed for FLOD) created about 32,000 triples, i.e., the FLOD-originated triples from 2,148,128 increased to 2,180,678.

Table 4: MarineTLO-based warehouses using OWLIM and Virtuoso

KB Part	# triples in OWLIM	# triples in Virtuoso
MarineTLO	277	58
FLOD	9,092,087	2,148,128
ECOSCOPE	170,980	84,184
WoRMS	70,174	9,552
FLOD-2-TLO mappings	180	15
ECOSCOPE-2-TLO mappings	205	11

[15] Exclusive Economic Zone

[16] Experiments done using a QuadCore Linux machine with 4 GB RAM with OWLIM version 4.2 and Virtuoso opensource version 6.1

WoRMS-2-TLO mappings	180	8
Total	10,822,758	2,241,956

To test query performance, we used queries provided by the iMarine partners. The average time in OWLIM ranged from 62ms to 8.8 seconds, while in Virtuoso from 31ms to 3.4 seconds. We observe that Virtuoso is faster despite the fact that OWLIM keeps everything in main memory, while Virtuoso does not necessarily do so. In general, performance depends on the capabilities of the adopted triplestore used (for a comparative analysis see [Haslhofer,2011]).

5.2.1 The contents of the `MarineTLO`-based warehouse(-s)

Based on the above results, we decided to use Virtuoso for the subsequent versions of the warehouse. Similarly to the different versions of the **`MarineTLO`**, we released 4 different version of the warehouse. Each version contained the corresponding **`MarineTLO`** version and the required schema mappings, in addition to the following:

- Version 1: contents from FLOD, ECOSCOPE and WoRMS, about the scientific name and predators of species.
- Version 2: contents from FLOD, ECOSCOPE, WoRMS and DBpedia, about the same concepts of Version 1 (i.e., scientific names and predators) plus authorship information of species.
- Version 3: contents from FLOD, ECOSCOPE, WoRMS, FishBase and DBpedia about the same concepts of Version 2 plus common names of species, information about ecosystems, countries, water areas, vessels, gears and EEZ. After the Version 3 release we released another version (named Version 3+) having the same contents with Version 3, however, we used multiple graphspaces for storing data coming from different sources. This allowed us to track easily the provenance of the information in the warehouse (e.g., the fact that "*yellowfin tuna*" is an English common name of the species `thunnus albacares` is derived from WoRMS and FishBase).
- Version 4: contents from FLOD, ECOSCOPE, WoRMS, FishBase and DBpedia about the same concepts of Version 3, containing also information about bibliographic citations and statistical indicators.

Figure 13 shows the differences between the 4 versions of the **`MarineTLO`**-based warehouse, in terms of the number of triples, species, main concepts and used sources. The first plot (A) shows how the number of species has been increased from 10 thousand (in the first version of the warehouse) to 53 thousand (in the fourth version). The second plot (B) depicts the increment in the size of the triplestore. Data are described in the warehouse as triples (in the form of <*subject*, *predicate*, *object*>), so the plot depicts the number of triples for the different versions. Plot (C) shows the different concepts (i.e. scientific names, predators, vessels, etc.) which are included in the different version of the **`MarineTLO`**-based warehouse and the last one (D) illustrates the number of the underlying sources which are exploited in each version.

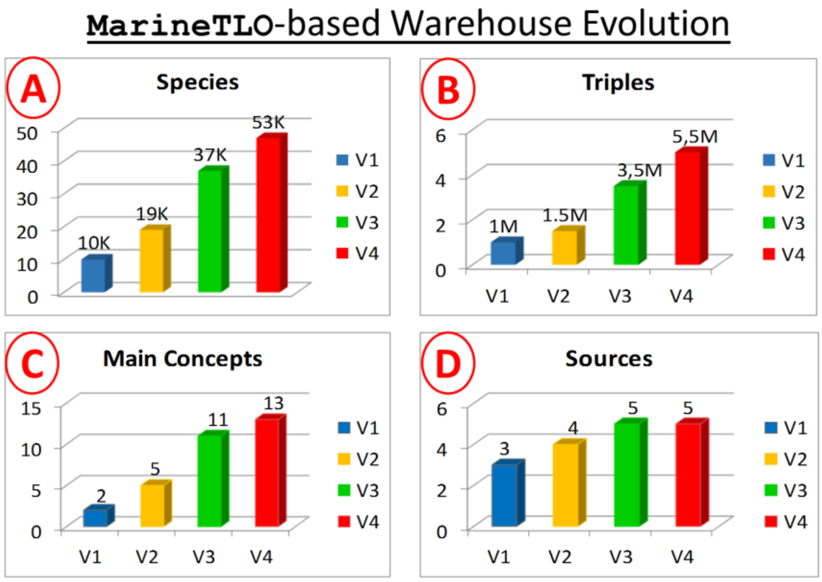


Figure 13: The evolution history of MarineTLO-based warehouse

5.3 Current uses of MarineTLO-based Warehouse

The **MarineTLO**-based warehouse is under constant evolution. At the time of writing, it contained information about 54 thousand species (i.e., scientific and common names, predators, bibliographic resources, ecosystems, water areas etc.). A SPARQL endpoint is available online[17]. Figure 14 shows the contents of the latest version of the **MarineTLO**-based warehouse.

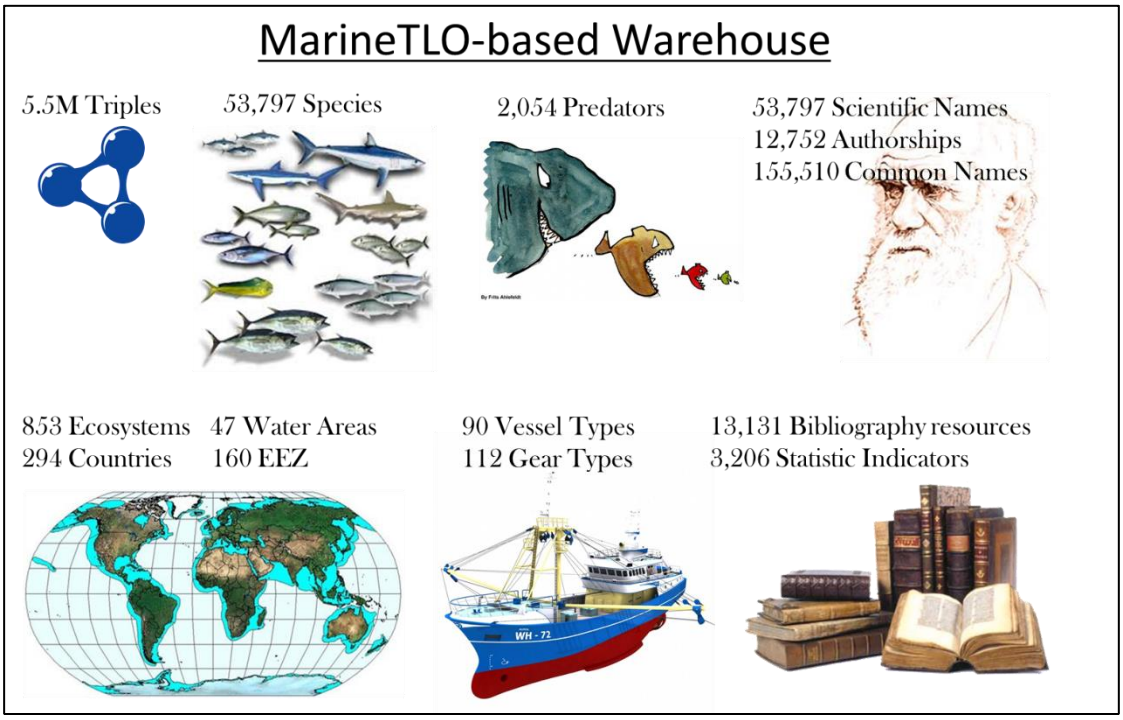


Figure 14: The contents of the MarineTLO-based warehouse (on July 2014)

[17] The warehouse can be accessed from https://i-marine.d4science.org/. Instructions for connecting and using is can be found at http://www.ics.forth.gr/isl/MarineTLO/files/AccessingMarineTLOBasedWarehouse.pdf

This warehouse is currently in use by the X-Search[18] system. Before building the **MarineTLO**-based warehouse, X-Search was exploiting FLOD as the underlying knowledge base and was able to detect no more than 11,000 species. Note also that for each species, the **MarineTLO**-based warehouse has in average about 30 properties, while in FLOD each species has in average only 6 properties. In addition, the **MarineTLO**-based warehouse contains about 200 distinct predicates that connect two URIs (contrary to the about 40 predicates of FLOD), allowing richer experience while browsing on the properties of an entity. The left part of Figure 15(i) depicts an example of (a part of) an entity card. An entity card is a popup window describing a resource (e.g., a species) which is displayed to the user on-demand (by clicking the small icon next to an entity name in Figure 3), offering entity exploration and browsing. In that figure, we divided the card into four groups, each one presenting information derived from different sources. Specifically, group A comes from DBpedia, B from FLOD, C from ECOSCOPE and D from WoRMS. Note that this information is derived at real-time (in less than one second).

Furthermore, the FactSheetGenerator (described in § 2.2) for using this warehouse is under development and will offer more elaborate information. Its current version focuses on tuna species and is called TunaAtlas[19]. An indicative screen of a prototype is given in Figure 15(iii).

Finally, we have developed (and currently improve) an Android application, called `Ichthys` that exploits the contents of the warehouse aiming to offer to end users information about marine species in a user friendly manner. Screen samples are shown in Figure 15(ii).

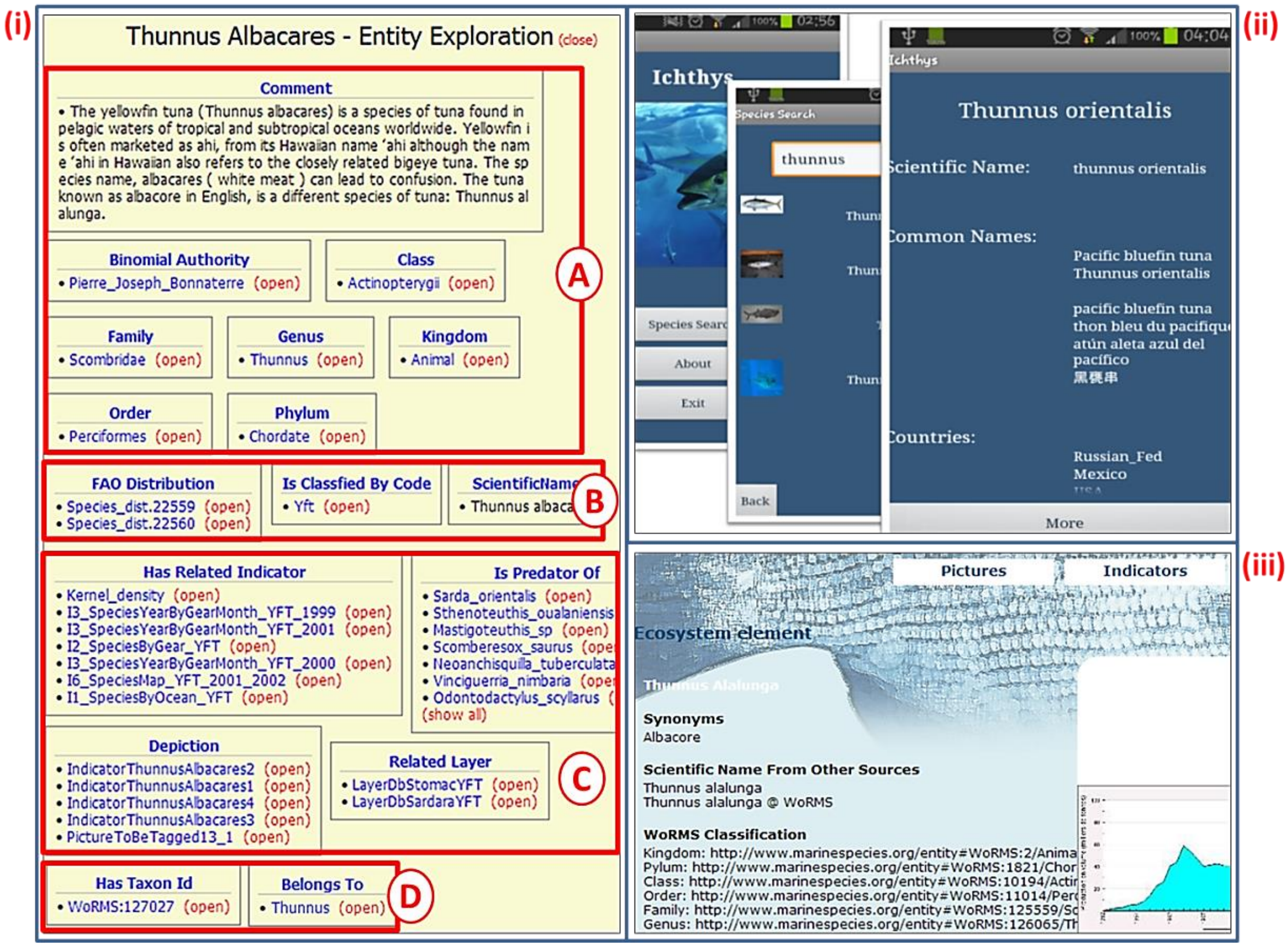
6 Concluding Remarks

In this paper, we described the design of a top level ontology for the marine domain, intended to satisfy the need for maintaining integrated sets of facts about marine species, and thus assisting ongoing research on biodiversity. The ontology offers a unified and coherent core model for schema mapping, which enables the formulation and answering of complex queries that cannot be answered by any individual source alone. We identified and described use cases and applications that exploit this ontology, and elaborated on the mappings that are required to build integrated warehouses. Finally, we discussed the realization of the mappings given the reasoning capabilities of the selected triplestore and evaluated the warehouse with respect to its completeness and its ability to answer the complex queries.

In the future, we plan to continue along the same lines and evolve **MarineTLO** by considering more sources and more competence queries, and to enhance the configurability of the workflow used for producing **MarineTLO**-based wareshouses.

To conclude, MarineTLO will also be exploited in the context of the LifeWatch Greece project[20], as the core underlying schema of the Lifewatch Greece infrastructures. Towards this end, it will be extended to cover also terrestrial and fresh water domains, microCT scanning processes, genetics, morphometric characteristics, and more.

[18] http://www.ics.forth.gr/isl/X-Search
[19] http://www.i-marine.eu/Content/About.aspx?id=f0fd33e9-b4bf-41b4-a746-46c0981913cc
[20] http://www.lifewatchgreece.eu/



Thunnus Albacares - Entity Exploration (close)

Comment
- The yellowfin tuna (Thunnus albacares) is a species of tuna found in pelagic waters of tropical and subtropical oceans worldwide. Yellowfin i s often marketed as ahi, from its Hawaiian name 'ahi although the nam e 'ahi in Hawaiian also refers to the closely related bigeye tuna. The sp ecies name, albacares (white meat) can lead to confusion. The tuna known as albacore in English, is a different species of tuna: Thunnus al alunga.

Binomial Authority
- Pierre_Joseph_Bonnaterre (open)

Class
- Actinopterygii (open)

Family
- Scombridae (open)

Genus
- Thunnus (open)

Kingdom
- Animal (open)

Order
- Perciformes (open)

Phylum
- Chordate (open)

FAO Distribution
- Species_dist.22559 (open)
- Species_dist.22560 (open)

Is Classfied By Code
- Yft (open)

ScientificName
- Thunnus albaca

Has Related Indicator
- Kernel_density (open)
- I3_SpeciesYearByGearMonth_YFT_1999 (open)
- I3_SpeciesYearByGearMonth_YFT_2001 (open)
- I2_SpeciesByGear_YFT (open)
- I3_SpeciesYearByGearMonth_YFT_2000 (open)
- I6_SpeciesMap_YFT_2001_2002 (open)
- I1_SpeciesByOcean_YFT (open)

Is Predator Of
- Sarda_orientalis (open)
- Sthenoteuthis_oualaniensis
- Mastigoteuthis_sp (open)
- Scomberesox_saurus (ope
- Neoanchisquilla_tuberculata
- Vinciguerria_nimbaria (ope
- Odontodactylus_scyllarus (
(show all)

Depiction
- IndicatorThunnusAlbacares2 (open)
- IndicatorThunnusAlbacares1 (open)
- IndicatorThunnusAlbacares4 (open)
- IndicatorThunnusAlbacares3 (open)
- PictureToBeTagged13_1 (open)

Related Layer
- LayerDbStomacYFT (open)
- LayerDbSardaraYFT (open)

Has Taxon Id
- WoRMS:127027 (open)

Belongs To
- Thunnus (open)

Ichthys

Species Search

thunnus

Thunn

Species Searc

About

Exit

Back

Ichthys

Thunnus orientalis

Scientific Name: thunnus orientalis

Common Names:
Pacific bluefin tuna
Thunnus orientalis

pacific bluefin tuna
thon bleu du pacifiqu
atún aleta azul del pacífico
黑鮪串

Countries:
Russian_Fed
Mexico
USA

More

Pictures

Indicators

Ecosystem element

Thunnus Alalunga

Synonyms
Albacore

Scientific Name From Other Sources
Thunnus alalunga
Thunnus alalunga @ WoRMS

WoRMS Classification
Kingdom: http://www.marinespecies.org/entity#WoRMS:2/Anima
Pylum: http://www.marinespecies.org/entity#WoRMS:1821/Chor
Class: http://www.marinespecies.org/entity#WoRMS:10194/Actin
Order: http://www.marinespecies.org/entity#WoRMS:11014/Perc
Family: http://www.marinespecies.org/entity#WoRMS:125559/Sc
Genus: http://www.marinespecies.org/entity#WoRMS:126065/Th

Figure 15: Usages of MarineTLO-based warehouse. (i) an entity exploration card displayed by XSearch for the species Thunnus Albacares, (ii) Screenshots from the Icthys Android application, (iii) the Tuna Atlas application

Acknowledgement: This work was partially supported by the project iMarine (FP7 Research Infrastructures, 2011-2014) and the LifeWatch Greece project (National Strategic Reference Framework, 2012-2015), and the European Union and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - Research Funding Program (Thales, Investing in knowledge society through the European Social Fund).

References

[Candela et al., 2008]: L. Candela, D. Castelli, P. Pagano, gCube: a service-oriented application framework on the grid. ERCIM News, 72 (2008), pp. 48–49 (October).

[Candela et al., 2014-1]: L. Candela, D. Castelli, G. Coro, L. Lelii, F. Mangiacrapa, V. Marioli, P. Pagano, An Infrastructure-oriented Approach for supporting Biodiversity Research. Ecological Informatics, Elsevier, 2014, doi: 10.1016/j.ecoinf.2014.07.006.

[Candela et al., 2014-2]: L. Candela, F. De Faveri, L. Lelli, F. Mangiacrapa, V. Marioli, P. Pagano. Accessing biodiversity databases: a domain specific query language and a unifying data model. Technical Report 2014-TR-26, Istituto di Scienza e Tecnologie dell'Informazione A. Faedo, CNR (2014).

[Colombo et al., 2009]: G. Colombo, D. Merico, Z. Nagy, F. De Paoli, M. Antoniotti, and G. Mauri. Ontological Modeling at a Domain Interface: Bridging Clinical and Biomolecular Knowledge. The Knowledge Engineering Review, Vol. 24:3, 205–224. 2009, Cambridge University Press, doi:10.1017/S0269888909990026.

[Doerr et al., 2003] :M. Doerr, J. Hunter, and C. Lagoze. Towards a core ontology for information integration. Journal of Digital Information, 4:2003, 2003.

[Fafalios et al., 2012]: P. Fafalios, I. Kitsos, Y. Marketakis, C. Baldassarre, M. Salampasis, and Y. Tzitzikas. Web searching with entity mining at query time. In Procs of the 5th Information Retrieval Facility Conference, 2012.

[Fafalios and Tzitzikas,2013]: P. Fafalios and Y. Tzitzikas. X-ENS: Semantic Enrichment of Web Search Results at Real-Time. In Procs of SIGIR'13, 2013.

[Gangemi et al., 2002] A. Gangemi, F. Fisseha, I. Pettman, and J. Keizer. Building an integrated formal ontology for semantic interoperability in the fishery domain. In Procs of ISWC'2002, 2002.

[Haslhofer et al., 2011]: B. Haslhofer, E. Momeni Roochi, B. Schandl, and S. Zander. Europeana RDF store report. 2011.

[Hitzler et al., 2009]: P. Hitzler, M. Krtzsch, B. Parsia, P. F. Patel-Schneider, and S. Rudolph. OWL 2 Web Ontology Language Primer. W3C Recommendation, World Wide Web Consortium, October 2009.

[Ibrahim and Pyster,2004]: L. Ibrahim and A. Pyster. A single model for process improvement. IT Professional, 6(3), May 2004.

[Madin et al., 2007]: J. Madin, S. Bowers, M. Schildhauer, S. Krivov, D. Pennington, and F. Villa. An ontology for describing and synthesizing ecological observation data. Ecological informatics, 2(3):279{296, 2007.

[Moreau et al., 2011]: L. Moreau, B. Clifford, J. Freire, J. Futrelle, Y. Gil, P. Groth, N. Kwasnikowska, S.Miles, P. Missier, J. Myers, B. Plale, Y. Simmhan, E. Stephan, J.V. den Bussche, The open provenance model core specification (v1.1), Future Generation Computer Systems 27 (6) (2011) 743–756. doi:10.1016/j.future.2010.07. 005. URL: http://www.sciencedirect.com/science/article/B6V06-50J9GPP-3/2/09d841ac888ed813ccc3cce84383ce27.

[Mosca et al., 2009]: A. Mosca, M. Palmonari, and F. Sartori. An upper-level Functional Design Ontology to Support Knowledge Management in SME-based E-Manufacturing of Mechanical Products. The Knowledge Engineering Review, Vol. 24:3, 265–285. & 2009, Cambridge University Press. doi:10.1017/S0269888909990063.

[Mountantonakis et al., 2014]: M. Mountantonakis, C. Allocca, P. Fafalios, N. Minadakis, Y. Marketakis, C. Lantzaki, and Y. Tzitzikas. Extending VoID for expressing the connectivity metrics of a semantic warehouse. In 1st International Workshop on Dataset Profiling & Federated Search for Linked Data (PROFILES’14), Anissaras, Crete, Greece, May 2014.

[Noessner et al., 2010]: J. Noessner, M. Niepert, C. Meilicke, and H. Stuckenschmidt. Leveraging terminological structure for object reconciliation. Procs of ESWC'10, 2010.

[Sacco and Tziztikas,2009]: G. M. Sacco and Y. Tzitzikas. Dynamic Taxonomies and Faceted Search: Theory, Practice, and Experience, volume 25. Springer, 2009.

[Simeoni et al., 2007]: F. Simeoni, L. Candela, G. Kakaletris, M. Sibeko, P. Pagano, G. Papanikos, P. Polydoras, Y. Ioannidis, D. Aarvaag, F. Crestani. A Grid-based Infrastructure for Distributed Retrieval. In Research and Advanced Technology for Digital Libraries, 11th European Conference, ECDL 2007, Budapest, Hungary, September 16-21, 2007, Proceedings, Springer-Verlag, 2007, 4675, 161-173

[TDWG,2004]: TDWG. Darwin core schema (version 1.3), a draft standard of the taxonomic database working group (tdwg), 2004.

[Tzitzikas et al., 2014-1]: Y. Tzitzikas, N. Minadakis, Y. Marketakis, P. Fafalios, C. Allocca, and M. Mountantonakis. Quantifying the connectivity of a semantic warehouse. In 4th International Workshop on Linked Web Data Management (LWDM’14), Athens, Greece, March’2014.

[Tzitzikas et al., 2014-2]Q:Y. Tzitzikas, N. Minadakis, Y. Marketakis, P. Fafalios, C. Allocca, M. Mountantonakis, and I. Zidianaki. MatWare: Constructing and exploiting domain specific warehouses by aggregating semantic data.In 11th Extended Semantic Web Conference (ESWC’14), Anissaras, Crete, Greece, May 2014.

[Wieczorek et al., 2012] J. Wieczorek, D. Bloom, R. Guralnick, S. Blum, M. Döring, R.D.T. Robertson, D. Vieglais. Darwin Core: an evolving community-developed biodiversity data standard. PLoS One, 7 (1) (2012).

[Wilson, 2009]: E. Wilson. Metadata for plant seeds: Taxonomy, standards, issues, and impact. Library Philosophy and Practice (e-journal), page 306, 2009.