Integrating Heterogeneous and Distributed Information about Marine Species through a Top Level Ontology

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 Candela[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

Abstract. One of the main characteristics of biodiversity data is its cross-disciplinary feature and the extremely broad range of data types, structures, and semantic concepts which encompasses. Moreover, biodiversity data, especially in the marine domain, is widely distributed, with few well-established repositories or standard protocols for their archiving, access, and retrieval. Our research aims at providing models and methods that allow integrating such information either for publishing it, browsing it, or querying it. For providing a valid and reliable knowledge ground for enabling semantic interoperability of marine data, in this paper we motivate a top level ontology, called `MarineTLO` that we have designed for this purpose, and discuss its use for creating `MarineTLO`-based warehouses in the context of a research infrastructure.

1 Introduction

Biodiversity data, especially in marine domain, is 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 how users could 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 are stored

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

in different sources (here FLOD, ECOSCOPE and WoRMS, 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.

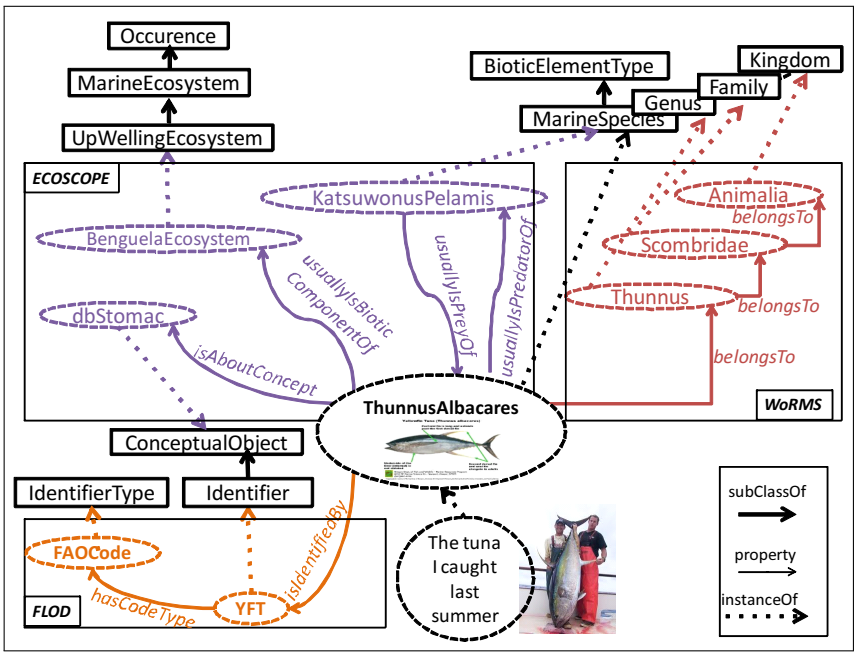

Fig. 1: Integrated information about `Thunnus albacares` from three sources

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: (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 three 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 are not answered by the underlying sources. To the best of our knowledge, there is not any other such warehouse. The rest of this paper is organized as follows. Section 2 discusses motivating application scenarios, Section 3 describes the proposed approach, and finally Section 4 concludes and identifies directions for future work and research.

2 Requirements and Motivating Scenarios

Here we describe the main underlying sources (§2.1), and then (§2.2) four motivating scenarios as came up by the organizations participating to iMarine.

2.1 Main Underlying Sources

• **Fisheries Linked Open Data (FLOD) RDF dataset.** FLOD (created and maintained by 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[2]. The FLOD content is exposed either via a public SPARQL endpoint[3] (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[4] offers a public SPARQL endpoint[5] for its knowledge base containing geographical data, pictures and information about marine ecosystems (specifically data about fishes, sharks, related persons, countries and organizations, harbours, vessels, etc.).
• **WoRMS.** The World Register of Marine Species[6] currently contains more than 200 thousands species, around 380 thousands species names including synonyms, and 470 thousands taxa (infraspecies to kingdoms).

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[7] 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

[2] Information from: http://www.fao.org/fishery/topic/18046/en
[3] http://www.fao.org/figis/flod/endpoint/sparql
[4] Institut de recherche pour le developpement (IRD), France (http://www.ird.fr/)
[5] http://ecoscopebc.mpl.ird.fr/joseki/ecoscope
[6] http://www.marinespecies.org/
[7] `http://www.ecoscopebc.ird.fr/`

knowledge stored in other sources (e.g. about commercial codes or taxonomic information) cannot be exploited. The approach that we will present in this paper can be exploited for advancing this application, i.e. for providing more complete semantic descriptions.

Fig. 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 [3, 4], proposed a method to enrich the classical (mainly keyword based) searching with *entity mining* that is performed at *query time*. In particular, 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 [10]. 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 indicated how semantic repositories can be exploited for specifying the entities of interest and for providing further information about the identified entities.

The current application within iMarine of this "semantic post-processing" service uses FLOD. Figure 3 shows a screendump of the results for the query `tuna` over a deployment as a portlet where the underlying system is *gcubeSearch* and the triplestore is FLOD. The approach presented in this paper can improve this service from various perspectives: more entities can be identified in the results, the system will be able to provide more complete information about the identified entities, etc.

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 such repositories:

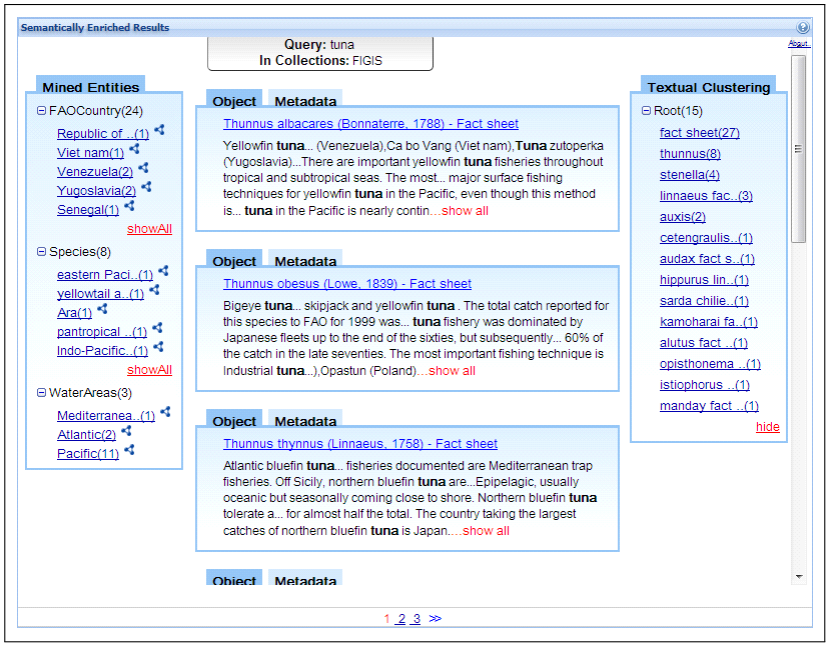

Fig. 3: Examples of semantic post-processing of search results within gcube as portlet

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 its 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, 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 [9]) 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 are *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 into one 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.

3 `MarineTLO`-based Integration

At first (§3.1) we describe `MarineTLO`, then (§3.2) we describe a discovery service (called `SDS`), and the process for creating `MarineTLO`-based descriptions, and finally (§3.3) the process for creating `MarineTLO`-based warehouses.

3.1 The ontology `MarineTLO`

`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)* data originating from distinct sources can be adequately mapped and integrated, as it happened for others and related domains [5],[2]. Figure 4 drafts the intended architecture of knowledge models.

Note that the adoption of a single and coherent core conceptual model has various benefits: *(a)* reduced effort for improving and evolving it since the focus is given on one model, rather than many [8], and *(b)* reduced effort for constructing

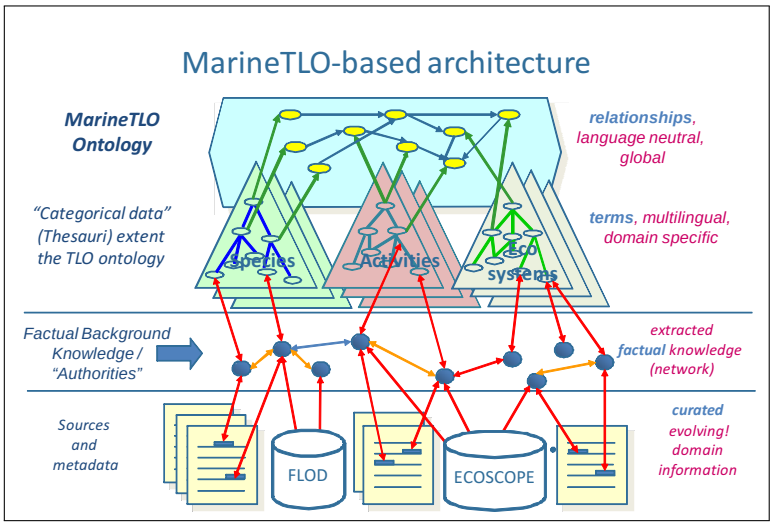

Fig. 4: MarineTLO-based architecture

mappings since this approach avoids the inevitable combinatorial explosion and complexities that results from pair-wise mappings between individual metadata formats and/or ontologies [2].

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 Ecoscope and FLOD, (ii) categories of classes (metaclasses) which support certain type of inference about classes in an analogous way as classes support certain types of inference about instances and enable the assignment of attribute values to a class. Also attention has been given to the design of MarineTLO for preserving *monotonicity*. Since the primary role of MarineTLO, is the meaningful integration of information in an Open World, 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 sibling concept under a common direct superclass.

Outcome. The current full version of MarineTLO contains 55 classes and 37 properties (its documentation is web accessible[8]). For the needs of the intended applications and the main underlying sources (i.e. FLOD, ECOSCOPE, WoRMS), only a subset of the full version is used and is further specialized. With the name "MarineTLO", we hereafter refer to this subset. Its current version (1.0) contains 47 classes and 8 properties. 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

[8] www.ics.forth.gr/isl/MarineTLO/documentation

generic queries. Figure 5 shows the metaclasses (and how they are organized in a `subClassOf` hierarchy), and 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. A short description of the role of each of the eight properties of `MarineTLO` follows (for reasons of space their domain and range is not discussed):

- `belongsTo`: it is used for the needs of taxonomic classification (species to genuses, genuses to families, and so on).
- `usuallyIsPredatorOf` (and its inverse `usuallyIsPreyOf`): they are superproperties for hosting the relations coming from ECOSCOPE.
- `usuallyFeedingOn`: it is a generalization (superproperty) of the relation `usuallyIsPredatorOf`.
- `hasIdentifierType:` it is used to link a species (e.g. `Thunnus Albacares`) with the types of codes of this species that are provided by various authorities (e.g. codes from FAO, IRD, WoRMS, etc).
- `isReferencedBy`: it allows stating that an information object (e.g. a picture) refers to a species.
- `usuallyIsComponentOf`: it is used to define the biotic constituent parts of a type of ecosystem (e.g. that `ThunnusAlbacares` is usually part of upwelling ecosystems).

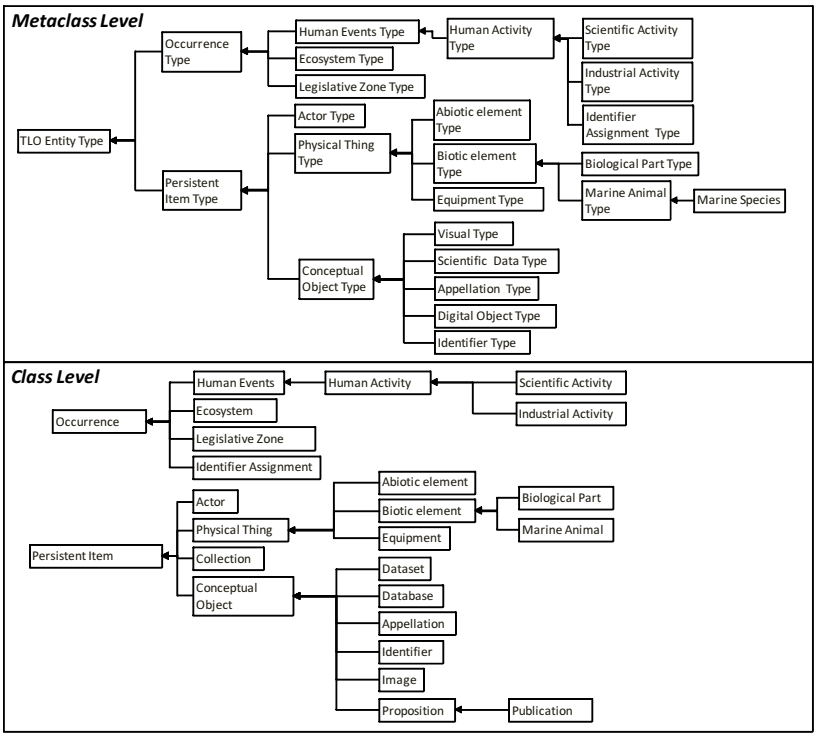

Fig. 5: The metaclasses and part of the classes of `MarineTLO`

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.

3.2 The Species Discovery Service (`SDS`) and its use for producing `MarineTLO`-based descriptions

The *Species Discovery Service*, for short `SDS`, under evolution in the context of iMarine and part of the gCube infrastructure [1], aims at offering an uniform access over different biodiversity repositories. It is a *plugin-based mediator service* for key biodiversity data sources that provides users with seamless access to both nomenclature data and species occurrences from the major information systems including GBIF and OBIS for occurrence data, Catalogue of Life, OBIS, Interim Register of Marine and Nonmarine Genera (IRMNG), ITIS, NCBI, and WoRMS for nomenclature data. We have implemented a tool that uses `SDS` API and transforms the fetched information into descriptions structured according to the `MarineTLO`. Its functionality is performed in two phases: the first takes as input a list of scientific names to be retrieved and the data sources that will be searched and submits the query to SDS. 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` through the used mappings. All the scientific names in the archive are classified under certain `MarineTLO` classes and the associations (w.r.t. their classification) are also added.

3.3 On constructing `MarineTLO`-based warehouses

We have been investigating the materialized (warehouse) approach described in section 2.2. Specifically we coded the `MarineTLO` ontology using OWL 2 [7] and set up a repository using two different triples stores which are described in the next section. Apart from `MarineTLO`, the repository contains the entire FLOD, the entire ECOSCOPE, and a part of WoRMS derived by running the process just described for 95 species, and the required mappings (between `MarineTLO` and FLOD, ECOSCOPE and WoRMS) which are described next.

Used Triplestores We have comparatively evaluated two different triplestores: OWLIM-Lite and Virtuoso. The first has been designed for medium data volumes (below 100 million statements). It contains a persistence layer, however reasoning and query evaluation are being performed entirely in main memory. On the other hand *OpenLink Virtuoso* supports *backward chaining* reasoning, meaning that it does not materialize all inferred facts, but computes them at query level. Its reasoner covers the related entailment rules of `rdfs:subClassOf` and `rdfs:subPropertyOf`. Practically this means that transitive relations (i.e. *subClassof*, *subPropertyOf*, etc.) are not physically stored in the knowledge base, but they are added to the result set at query answering.

Mappings In general what we call *mapping* comprises: extensions to the metaschema, extensions to the schema, `rdfs:subClassOf` and `rdf:subPropertyOf` relationships between the elements of `MarineTLO` and the schema at hand, plus some inference rules. Below we sketch the defined mappings. For instance, the *ECOSCOPE2TLO mapping* consists of `subClassOf` and `subPropertyOf` like those shown in Figure 6. The *WORMS2TLO mapping* contains analogous relationships. 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 do one step further and look at its code. For instance, we can distinguish *FAOSpecies* as follows: $FAOSpecies = \{\ x \mid CodedEntity(x)\ and\ (\exists\ y\ isClassifiedByCode(x, y)\ and\ SpeciesCode(y))\}$. 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 a SPARQL INSERT query.

```
(tlo:EcoscopeSpecies, rdfs:subClassOf, tlo:TLOSpecies)
(ecoscope:fish, rdfs:subClassOf, tlo:EcoscopeSpecies)
(ecoscope:is_Predator_Of, rdfs:subPropertyOf,
tlo:usuallyisPredatorOf) (ecoscope:is_Prey_Of,  rdfs:subPropertyOf,
tlo:usuallyisPreyOf) (ecoscope:bioticComponentOf,
rdfs:subPropertyOf, tlo:usuallyisComponentOf)
(ecoscope:used_data_source, rdfs:subPropertyOf, tlo:isReferencedBy)
```

Fig. 6: Mapping ECOSCOPE2TLO

Comparison of the two repositories

Number of Triples and Loading Times. Table 1 shows the sizes in triples of the contents of the OWLIM and Virtuoso repositories. The first contains in total 10.8 millions 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 here takes 4h and 20 minutes[9]. 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.

Query Performance. To test query performance, we used queries provided by the iMarine partners (more below). The average time in OWLIM was ranging 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 [6]).

Evaluation For evaluating the stucturing of `MarineTLO`, and the process used for creating the `MarineTLO`-based repository, we had to investigate whether they

[9] Experiments done using a QuadCore linux machine with 4GB RAM.

Table 1: 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
FLOD2TLO mapping	180	15
ECOSCOPE2TLO mapping	205	11
WORMS2TLO mapping	180	8
TOTAL	10,822,758	2,241,956

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 2 shows *some* indicative and fundamental ones. The columns at the right show which of them are answerable by the underlying sources (fully or partially). The real competence queries include queries that combine more than one of the listed queries, e.g. *"I want the biotic types and the identifiers of the predators or competitors of the x species"*. Such queries cannot be answered by any particular source, and this is the concrete evidence of the benefits offered by the integrated model.

Table 2: Basic Queries

Query		**Ability to answer by**		
For the scientific name of a species, find:		*FLOD*	*Ecoscope*	*WoRMS*
i	its identifier in the involved sources (e.g. FLOD codes, ECOSCOPE codes, WoRMS Id)	partial	partial	partial
ii	its WoRMS classification			full
iii	its references/images/db		full	
iv	its biotic type		partial	full
v	its predators		full	
vi	its competitors		full	

3.4 Lessons Learned and Next Steps

In future (until April 2014) we plan to continue along the same lines and evolve MarineTLO by considering more sources and more competence queries, and enhancing the configurability of the workflow used for producing MarineTLO-based wareshouses. Another task that we do in parallel is the inspection of the repository for detecting the missing connections that are required for satisfying the needs of the competence queries. We currently use matching tools like SILK[10] for creating the missing relationships. The MarineTLO-based warehouse is under constant evolution. Today it contains information about 18,500 marine species. Apart from the three main sources, it currently includes information from dbpedia and a SPARQL endpoint is publicly available[11] and it is used

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

[11] http://virtuoso.i-marine.d4science.org:8890/sparql, also browsable through http://62.217.127.213:8890/fct/.

by various search services[12]. From this activity we have observed that the data fetched from the sources are in many cases problematic (consistency problems, duplicates, wrong values), and placing them together in a warehouse makes easier the identification of such errors. Moreover, the availability of the warehouse enables defining `sameAs` connections by exploiting transitively induced equivalences.

4 Concluding Remarks

To tackle the need for having integrated sets of facts about marine species, and thus to assist research about species and biodiversity, we have described a top-level ontology for that domain. It provides a unified and coherent core model for schema mapping which enables formulating and answering queries which cannot be answered by any individual source. We have identified and described particular use cases and applications that exploit this ontology, and have focused on the mappings that are required for building integrated warehouses. We discussed the realization of the mappings depending on the reasoning capabilities of the selected triplestore and we evaluated the warehouse as regards its completeness and its ability to answer queries which are not answered by the underlying sources.

Acknowledgement This work was partially supported by the ongoing project *iMarine* (FP7 Research Infrastructures, 2011-2014).

References

1. L. Candela, D. Castelli, and P. Pagano. Managing big data through hybrid data infrastructures. *ERCIM News*, (89):37–38, 2012.
2. M. Doerr, J. Hunter, and C. Lagoze. Towards a core ontology for information integration. *Journal of Digital Information*, 4:2003, 2003.
3. 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.
4. P. Fafalios and Y. Tzitzikas. X-ENS: Semantic Enrichment of Web Search Results at Real-Time. In *Procs of SIGIR'13*, 2013.
5. 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.
6. B. Haslhofer, E. Momeni Roochi, B. Schandl, and S. Zander. Europeana RDF store report. 2011.
7. 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.
8. L. Ibrahim and A. Pyster. A single model for process improvement. *IT Professional*, 6(3), May 2004.
9. J. Noessner, M. Niepert, C. Meilicke, and H. Stuckenschmidt. Leveraging terminological structure for object reconciliation. *Procs of ESWC'10*, 2010.
10. G. M. Sacco and Y. Tzitzikas. *Dynamic Taxonomies and Faceted Search: Theory, Practice, and Experience*, volume 25. Springer, 2009.

[12] Including `http://62.217.127.118/x-search-fao/`.