A Workflow for Supporting the Evolution Requirements of RDF-based Semantic Warehouses

Yannis Marketakis*, Yannis Tzitzikas

Institute of Computer Science, FORTH-ICS &
Computer Science Department, University of Crete Heraklion, Greece
E-mails: {marketak, tzitzik}@ics.forth.gr
* Corresponding author

Aureliano Gentile, Bracken van Niekerk, Marc Taconet

Food and Agriculture Organization of the United Nations
Rome, Italy
E-mails: {aureliano.gentile, bracken.vanniekerk, marc.taconet}@fao.org

Abstract: Semantic data integration aims to exploit heterogeneous pieces of similar or complementary information for enabling integrated browsing and querying services. A quite common approach is the transformation from the original sources with respect to a common graph-based data model and the construction of a global semantic warehouse. The main problem is the periodic refreshment of the warehouse, as the contents from the data-sources change. This is a challenging requirement, not only because the transformations that were used for constructing the warehouse can be invalidated, but also because additional information may have been added in the semantic warehouse, which needs to be preserved after every reconstruction. In this paper, we focus on this particular problem using a semantic warehouse that integrates data about stocks and fisheries from various information systems, we detail the requirements related to the evolution of semantic warehouses, and propose a workflow for tackling them.

Keywords: Semantic Warehouse; Evolution; Semantic Data Integration; Preserve Updates; Refresh Workflow

1 Introduction

The Web of Data contains thousands of RDF datasets available online (see Mountantonakis & Tzitzikas (2019) for a recent survey), including cross-domain Knowledge Bases, such as DBpedia and Wikidata, domain-specific repositories, such as WarSampo (Hyvönen et al. 2016), DrugBank (Wishart et al. 2018), ORKG (Jaradeh et al. 2019), life science related datasets (Reis et al. 2019) and recently COVID-19 related datasets (Wang et al. 2020, R. Gazzotti 2020), as well as Markup data through `schema.org`. One important category of domain specific semantic repositories, are semantic warehouses, produced by integrating various evolving datasets. Such warehouses aim at offering a unified view of the data and enabling the answering of queries which cannot be answered by the individual datasets. However, such semantic warehouses have to be refreshed periodically, because the underlying datasets evolve. This is an issue, especially if the structure of the original data sources change as they evolve, since this invalidates the schema mappings that were used for transforming them and constructing the semantic warehouse. Another issue is raised if the semantic warehouse contains updated information that is not available from the original data sources, e.g. manual updates or additions carried out solely in the semantic warehouse. Although, semantic warehouses are usually constructed as read-only resources, sometimes updating them directly is required, e.g. for cleaning or normalizing data, or adding manually specific resources according to the needs. Dealing with those updates is quite challenging, because in most cases the updated information is not reflected in the original data sources. As a result, this information is a valuable piece that should be preserved and made available in a consistent way after every re-construction of the semantic warehouse.

In this paper we focus on that particular problem. We study this problem in a real setting, specifically on the Global Record of Stocks and Fisheries (Tzitzikas et al. 2019), for short GRSF, a semantic warehouse that integrates data about stocks and fisheries from various information systems, under the auspices of UN FAO. In brief, GRSF is capable of hosting the

corresponding information categorized into uniquely and globally identifiable records. Moreover, the construction of GRSF does not invalidate the process being followed so far, in the sense that the organizations that maintain the original data are expected to continue to play their key role in collecting and exposing them. In fact, GRSF does not generate new data, rather it collates information coming from the different database sources, facilitating the discovery of inventoried stocks and fisheries arranged into distinct domains.

Although, GRSF is constructed by collating information from other data sources, it is not meant to be used as a read-only data source. After its initial construction, GRSF is being assessed by GRSF administrators who can edit particular information, like for example the short name of a record, update its connections, suggest merging multiple records into a new one (more about the merging process is given in §2.1), or even provide narrative annotations. The assessment process might result in approving a record, which will make it accessible to a wider audience through a publicly accessible URL. In general, GRSF URLs are immutable, and especially if a GRSF record becomes public then its URL should become permanent as well. The challenge when refreshing it, is that we want to be able to preserve the immutable URLs of the catalogue, especially the public ones. In addition, we want to preserve all the updates carried out from GRSF administrators, since their updates are stored in GRSF and are not directly reflected to the original sources. To do this, we need to identify records, and so we exploit their identifiers at different levels.

This paper provides an extended version of the approach described in Marketakis et al. (2021). In comparison to the previous work, in this paper we provide more details about the steps of the refresh workflow, we describe which are the benefits that each source is gaining from the curated semantic warehouse, and we attempt to generalize the problem by widening the evolution requirements to demonstrate its applicability to other domains.

The rest of this paper is organized as follows: Section 2 discusses background and requirements, Section 3 describes related work, Section 4 details our approach, Section 5 reports our experience on the implementation, Section 6 discusses the applicability of this method and related issues, and finally Section 7 concludes the paper and elaborates with future work and research.

2 Context

This section provides background information about the domain-specific warehouse GRSF (in §2.1), the activities carried out for complying GRSF with standards (in §2.2), and then discuss the evolution-related requirements (in §2.3).

2.1 Background: GRSF

The design and the initial implementation of the Global Record of Stocks and Fisheries have been initiated in the context of the H2020 EU Project BlueBRIDGE[a], and is currently maintained in the context of the ongoing H2020 EU Project BlueCloud[b].

It semantically integrates data from three different data sources, owned by different stakeholders, in the GRSF knowledge base (KB), and then exposes them through a catalogue of a Virtual Research Environment (VRE), operated on top of D4Science infrastructure (Assante et al. 2019). These data sources are: (a) Fisheries and Resources Monitoring System (FIRMS)[c], (b) RAM Legacy Stock Assessment database[d], and (c) FishSource[e]. They contain complementary information (both conceptually and geographically); FIRMS is mostly reporting about stocks and fisheries at regional level, RAM is reporting stocks at national or subnational level, and FishSource is more focused on the fishing activities.

GRSF is organized in units of information called *stocks* and *fisheries* records. Each record is composed of several fields to accommodate the incoming information and data. The fields can be functionally divided into *time-independent* and *time-dependent*. The former consists of identification and descriptive information that can be used for uniquely identifying a record, while the latter contains indicators which are modeled as dimensions. For example for the case of stock records such dimensions are the abundance levels, fishing pressure, biomasses, while for fishery records they are catches and landings indicators.

The process for constructing the initial version of GRSF is described in (Tzitzikas et al. 2019). Figure 1 shows a Use Case diagram depicting the different actors that are involved, as well as the various use cases. In general there are three types of users: (a) *Maintainers* that are responsible for constructing and maintaining GRSF KB, as well as publishing the concrete records from the KB to the VRE catalogues. They are the technical experts carrying out the semantic data integration from the original data sources. (b) *Administrators*, that are responsible for assessing information of GRSF records through the VRE catalogues, in order to validate their contents, as well as for spotting new potential merges of records. They are marine experts familiar with the terminologies, standards, and processes for assessing stocks and fisheries. Upon successful assessment they approve GRSF records which become available to external users. (c) *External users* for querying and browsing it. To ease understanding, Table 1 provides some background

[a] BlueBRIDGE (H2020-BG-2019-1), GA no 675680
[b] BlueCloud (H2020-EU.3.2.5.1), GA no: 862409
[c] `http://firms.fao.org/firms/en`
[d] `https://www.ramlegacy.org/`
[e] `https://www.fishsource.org/`

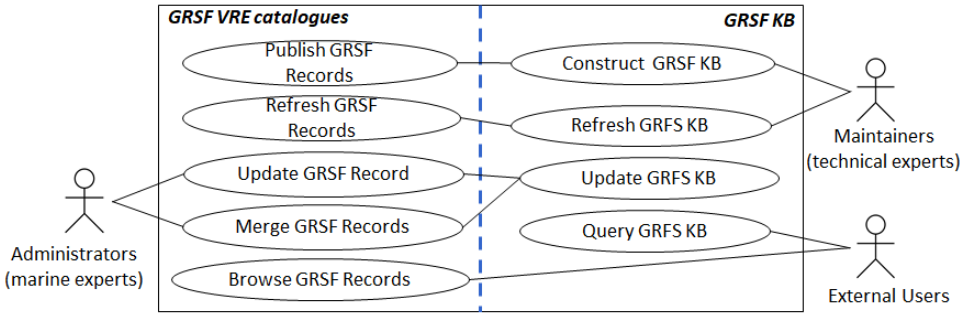

Figure 1 Use Case diagram of GRSF activities and involved persons

Term	Description
Source Record	A record that has been derived by transforming its original contents, with respect to a core ontology, specifically MarineTLO (Tzitzikas et al. 2016). For each record harvested from the original sources, we create a single source record and ingest it in GRSF KB.
GRSF Record	A new record that has been constructed taking information from one or more source records. GRSF records are described in a similar manner with source records (i.e. as ontology-based descriptions), however during their construction they adopt GRSF rules, and use global standard classification as much as possible, generate new attributes (e.g. semantic ID), flags, citations, etc. GRSF records can be the result of a merging or dissection activity (described below).
Semantic ID	They are *identifiers* assigned to GRSF records that are generated following a particular pattern and are meant to be both human and machine understandable. They are called semantic identifiers in the sense that their values allow identifying several aspects of a record. The identifier is a concatenation of a set of predefined fields of the record in a particular form. To keep them as short as possible it has been decided to rely on standard values or abbreviations whenever applicable. Each abbreviation is accompanied with the thesaurus or standard scheme that defines it. For GRSF stocks the fields that are used are: (1) species and (2) water areas (e.g. `ASFIS:SWO+FAO:34`). For GRSF fisheries the fields that are used are: (1) species, (2) water areas, (3) management authorities, (4) fishing gears, and (5) flag states (e.g. `ASFIS:COD+FAO:21+authority:INT:NAFO+ISSCFG:03.1.2+ ISO3:CAN`). The symbol '+' is used as a separator for concatenating the different fields of the semantic ID.
Merge	A process ensuring that source records from different sources having exactly the same attributes that are used for identification, are both used for constructing a single GRSF Stock record. The same attributes that are used for constructing the Semantic ID, are used for identifying records. An example is shown in Figure 2.
Dissect	A process applied to aggregated source fishery records so that they will construct concrete GRSF fishery records compliant with the standards. The process is applied on particular fields of the aggregated record (i.e. species, fishing gears, and flag states) so that the constructed GRSF record is uniquely described and suitable for traceability purposes. Considering that the source fishery record example contains two different species, the dissection process produces two distinct GRSF fishery records. An example is shown in Figure 2.
Approved Record	After their construction GRSF records, appear with status *pending*. Once they are assessed from GRSF administrators, they can be approved and as a result their status is changed to *approved*. Approved records are then made publicly available.
Dominant Record	When two or more source stock records are merged for constructing a concrete GRSF record, is it necessary to identify which of them is the dominant record. It is used for avoiding conflicts with the selection of time-independent values from the records that are used for merging (e.g. which short name, or geo polygon to use for the merged record).

Table 1 Explanation of the Terminology in GRSF

information about the terminologies of GRSF that are used in the sequel.

Figure 3 shows the different activities that are carried out. Initially, information from the data sources is harvested and transformed with respect to the core ontology that is used. Afterwards, the transformed data are conformed with respect to the adopted standards (more about this in §2.2) and finaly ingested into the GRSF KB, as source records, which are afterwards used for constructing the GRSF records, based on a set of well defined GRSF rules and after applying the corresponding activities (i.e. merging and dissection). Both the source

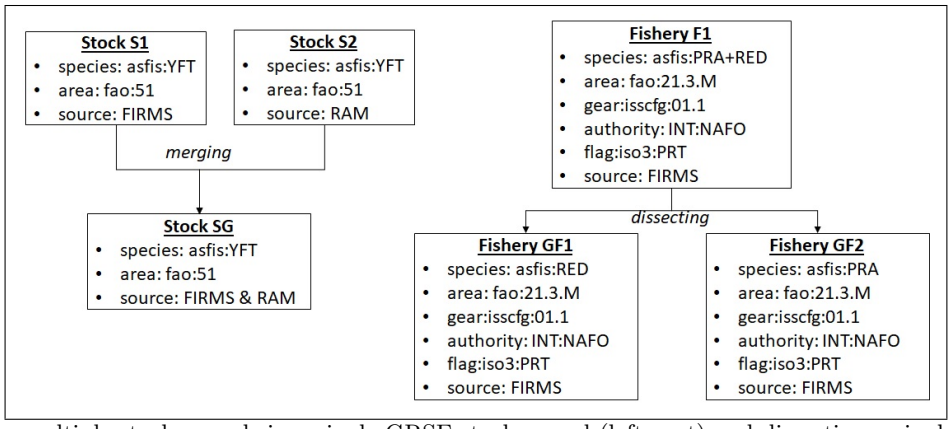

Figure 2 Merging multiple stock records in a single GRSF stock record (left part) and dissecting a single fishery record in multiple GRSF fishery records (right part)

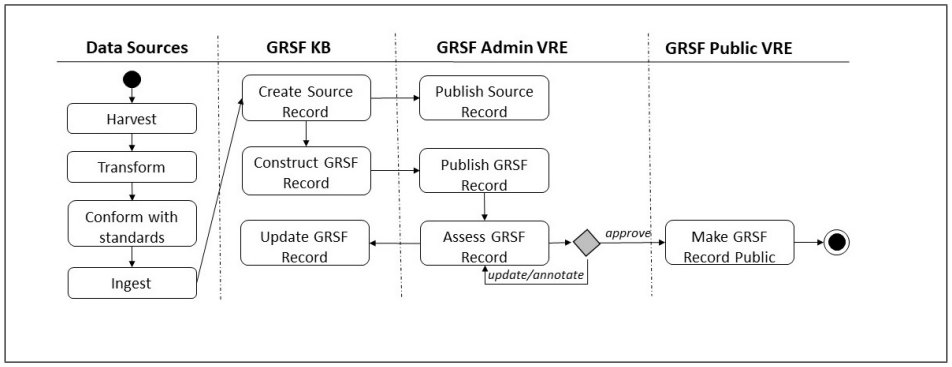

Figure 3 The process of constructing, publishing and assessing GRSF records

and GRSF records are published in the catalogue of a VRE. The former for provenance reasons and the latter for inspection and validation from GRSF administrators. When a GRSF record is approved, it becomes publicly available by replicating its contents in a public VRE.

2.2 Compliance with standards

The key for interoperability is standardization, and GRSF is constructed by exploiting international standards as much as possible. The adoption of standards ensures that information in the GRSF records are well defined and clearly understood, omitting potential ambiguity in the interpretation of the records. The exploited standards have been agreed, between GRSF maintainers and representatives of the data source providers that currently contribute their records to GRSF. In addition, many of the adopted standards, that are described below, are already used by the original data sources. Moreover, we should mention that, in some cases, there are more than one standards that are accepted for a particular resource type; for example for describing areas, we could use FAO codes, GFCM codes, Large Marine Ecosystems (LMEs) codes, etc. In this case, we apply a prioritization over the different standards that can be used for describing a piece of information. Furthermore, in some cases we can exploit mappings between different standards, in order to transform values from a standard with lower priority to a higher one. Below we describe the standard schemes that are exploited in GRSF.

• **Marine species:** There are various ways for identifying a marine species; non-expert users use their common names (e.g. yellowfin tuna), however it is not the best alternative since there are multiple common names for a particular species (in several different languages). One alternative for identifying species is their scientific name (e.g. Thunnus albacares), which is composed of two parts; the first being the genus name and the second the specific epithet. Another alternative for identifying species is their codes. FAO 3Alpha codes have been introduced by ASFIS, and consist of three letters that uniquely identify the species. In most of the cases, the codes have been derived either from the scientific name of the species, or by their common name in English (e.g. YFT is the FAO 3Alpha code for yellowfin tuna). In all other cases, the three letters are assigned at random. Another coding standard that is accepted is the APHIAID that provides a numeric code of the marine species. For the case of identifying marine species, ASFIS code is in the top of the priority, APHIAID follows, and the last in the list is the scientific name of the species.

• **Water areas:** Similarly to marine species, water areas can have commonly used names. However they are not adequate for identifying the area itself, since the boundaries of the area are not clearly defined. A more

accurate method is to describe them using polygons that are formulated using a list of geographic coordinates. A polygon is an accurate description of an area, since it can take any shape. A simpler abstraction is to use bounding boxes for modeling a water area. Apart from the above that can be used for visualizing records in a map, or identifying adjacent and/or overlapping records based on their geographic coverage, there is a set of standard coding systems that can be used for identifying them. One of them is FAO major fishing areas for statistical purposes[f] that provides an hierarchival coding system for indetifying water areas (e.g. the aegean sea has the FAO water area code 37.3.1). The exclusive economic zones is another coding standard that can be used[g]. In this case water areas are identified using the ISO3 code of the corresponding country. The General Fisheries Commission for the Mediterranean (GFCM) geographical sub-areas[h] is another alternative that can be used. For the case of areas described using GFCM codes, we also have the corresponding mappings to FAO codes. Large Marine Ecosystems (LMEs)[i] are wide areas of ocean space along the Earth's continental margins. Marine Regions provides a standardized list of georeferenced marine place names and marine areas in the form of MRGID[j].

• **Countries:** Countries can be described using their ISO3 codes[k]. These codes are composed of three letters and represent countries, dependent territories and special areas of geographical interest (e.g. the ISO Alpha-3 code for Greece is GRC).

• **Fishing Gear:** The Coordinating Working Party on fishery statistics (CWP)[l] provides a mechanism to coordinate fishery statistical programmes of regional fishery bodies and other inter-governmental organizations with a remit of fishery statistics. CWP adopted in 1980 a labeling and classification standard for fishing gears that led to the creation of the International Standard Statistical Classification of Fishing Gears (ISSCFG). The standard assigns an acronym and a classification code that can be used for identifying gears of the same type. For example portable lift nets are identified using the acronym LNP, while boat-operated lift nets are identified using the acronym LNB. The former has the classification code "05.1.0" while the latter has the code "05.2.0". The common prefix of the classification codes (e.g. "05") allow us identifying that they are similar types of fishing gears, in this case lift nets.

2.3 Evolution Requirements

The key requirements for supporting the evolution of GRSF are illustrated in an abstract form in Figure 4. The bottom part of the figure shows the original data sources and their evolution over time, while the upper part shows the corresponding evolution at GRSF level. In more details the key requirements are:

• (R1): *Refresh* the contents of GRSF with up-to-date information from the underlying sources for updating all the time-dependent information, as well as bringing potential fixes in the original records in GRSF.

• (R2): *Remove obsolete records* from GRSF and VRE catalogues: If their status is approved, then instead of removing them, change their status to *archived* and archive them in the VRE catalogue with a proper annotation message.

• (R3): *Preserve the immutable URLs* that are generated for GRSF records when they are published in VRE catalogues. These URLs should be preserved (instead of generating new ones) to avoid the creation of broken links.

• (R4): *Maintain* all the updates that have been carried out in GRSF records from GRSF administrators. These updates are performed in GRSF KB and are not applied back to the data sources (e.g. an update in the name of a record).

• (R5): *Maintain all the annotations* made by GRSF admininstrators to GRSF records (annotations are small narratives describing their observations during the assessment of the records).

• (R6): *Preserve all the merges* that are used for constructing GRSF records. Although GRSF merges are applied using a set of well-defined rules (as described in §2.1), GRSF administrators can propose and apply the merging of records manually. Since the latter might not be re-producable it is important to preserve them when refreshing GRSF.

3 Related Work and Novelty

There are several works that deal with the problem of evolution in ontology-based datasets and access in general. The problem of comparing two RDF datasets is discussed in Zeginis et al. (2011), while matching methods for RDF blank nodes to facilitate the comparison and to reduce the delta between two RDF datasets are analyzed in Lantzaki et al. (2017). A survey for ontology evolution is given in Flouris et al. (2008). The problem of query answering in mediators (virtual integration systems) under evolving ontologies without recreating mappings between the mediator and the underlying sources is studied in Kondylakis & Plexousakis (2013) where query re-writing methods are proposed. Note that in our case we tackle a different scenario: the sources evolve, not the ontology. The losses of specificity of ontology-based descriptions, when such descriptions are migrated to newer versions of the ontology have been studied in Tzitzikas et al. (2013).

There are various methods that focus on monitoring the "health" of various RDF-based systems, i.e. Käfer et al. (2013) focus on monitoring Linked Data over

[f] http://www.fao.org/fishery/area/search/en
[g] https://www.marineregions.org/eezmapper.php
[h] http://www.fao.org/gfcm/data/maps/gsas
[i] https://www.lmehub.net/
[j] https://www.marineregions.org/mrgid.php
[k] https://www.iban.com/country-codes
[l] http://www.fao.org/cwp-on-fishery-statistics/handbook

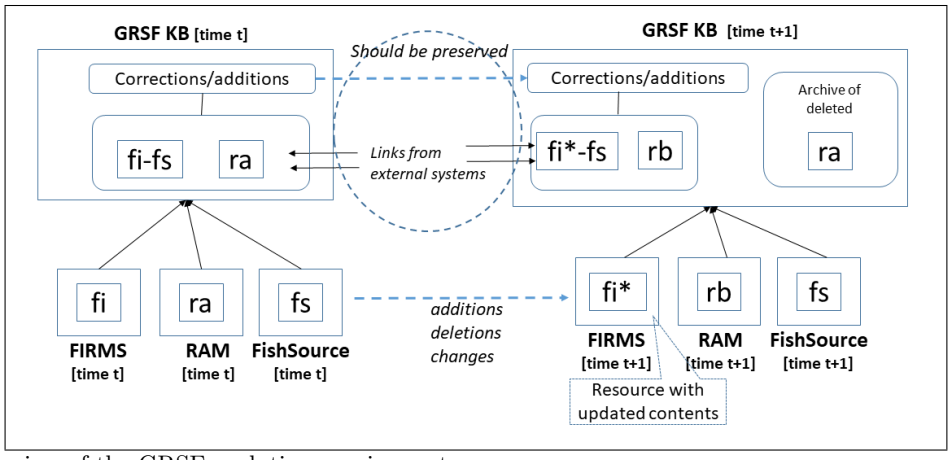

Figure 4 An overview of the GRSF evolution requirements

a specific period of time, Dividino et al. (2014) focus on measuring the dynamics of a specific RDF dataset, Roussakis et al. (2015) propose a framework that identifies, analyses and understands such dynamics, while Mountantonakis et al. (2016) focus on monitoring data connectivity in the context of a semantic warehouse over time,

SPARQLES (Vandenbussche et al. 2017) and *SpEnD* (Yumusak et al. 2017) focus on the monitoring of public SPARQL endpoints, *DyKOSMap* framework (Dos Reis et al. 2015) adapts the mappings of Knowledge Organization Systems, as the data are modified over time.

The work from Reis et al. (2019) is the one closest to our work. In that paper, the authors analyze the way change operations in RDF repositories correlate to changes observed in links. They investigated the behaviour of links in terms of complex changes (e.g. modification of triples) and simple ones (e.g. addition or removal of links). Compared to this work, and for tackling the GRSF requirements, in our work we focus on identifying and analyzing the evolution of each concrete *record* which is part of the GRSF dataset. Therefore instead of analyzing the evolution in terms of triples, we do it in terms of a collection of triples (e.g. a record), i.e. to an application-specific abstraction. Furthermore, we exploit the semantics of the links of a record by classifying them in different categories. For example, triples describing identifiers or URLs are classified as immutable and are not subject to change, while links pointing to time-dependent information are frequently updated. In addition, in our work we deal with the requirement of preserving manually provided information and various several human-provided updates and activities in the dataset, during its evolution. Finally, a recent survey on link maintenance for integrity in linked open data evolution is given in Regino et al. (2021).

4 An Approach for Semantic Warehouse Evolution

In §4.1 we elaborate on the identification of resources, while in §4.2 we detail the GRSF refresh workflow.

4.1 Uniquely Identifying Sources

Before actually refreshing information in GRSF KB, it is required to identify and map the appropriate information from the source databases, with information in the VRE catalogues and the GRSF KB. To do so, we will rely on identifiers for these records. The main problem, however, is raised from the fact that although data had identifiers assigned to them from their original sources, they were valid only within the scope of each particular source. As they have been integrated they were assigned a new identifier (i.e. in GRSF KB), and as they have been published in the VRE catalogues they have been assigned additional identifiers (i.e. in VRE catalogues). As regards the latter, it is a mandatory addition due to the different technologies that are used for GRSF. We could distinguish the identifiers in three distinct groups:

Data source identifiers. They are identifiers assigned to each record from the stakeholders of each source. If r denotes a record, let use $r.sourceID$ to denote its identifier in a source. For the cases of FIRMS and FishSource, they are short numbers (e.g. 10086), while for the case of RAM they are codes produced from the record details (e.g. PHFLOUNNHOKK). Furthermore, the first two sources have their records publicly available, through their identifiers with a resolvable URL representation (e.g. http://firms.fao.org/firms/resource/10089/en, https://www.fishsource.org/stock_page/1134).

GRSF KB identifiers. After the data have been harvested, they are transformed and ingested in GRSF KB. During the transformation they are assigned URIs (Uniform Resource Identifier), which are generated, by applying hashing over the data source identifier of the corresponding record, i.e. we could write $r.URI = hash(r.sourceID)$. This guarantees the uniqueness

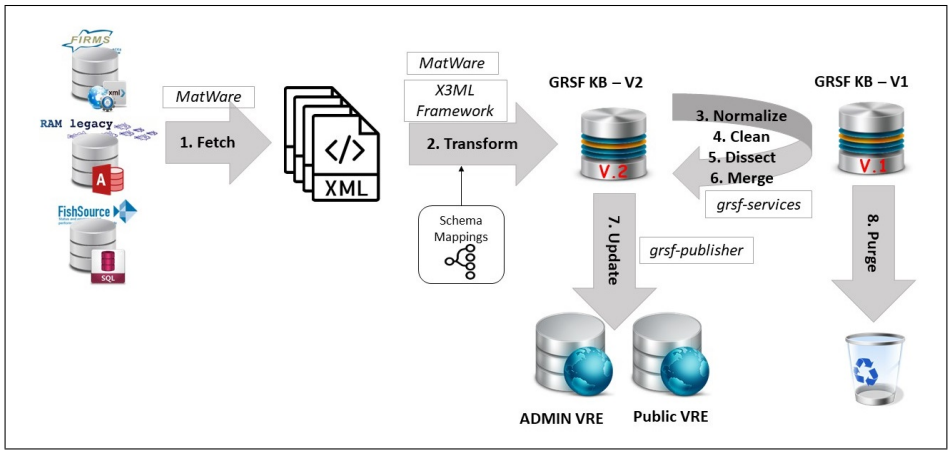

Figure 5 The workflow for refreshing GRSF, while preserving particular information from the previous version

of the URIs and avoids connecting irrelevant entities. Obviously, the data source identifiers are stored in GRSF KB, as well. For source records, URIs are generated based on the hashing described above, while for GRSF records a unique random URI is generated.

GRSF VRE catalogue identifiers. All the records from the GRSF KB, are published in the VRE catalogue, which enables their validation and assessment from GRSF Administrators. After publishing them in the catalogue, they are assigned a resolvable URL. The generated URL, denoted by $r.catalogID$, is stored in GRSF KB. These URLs are used for disseminating records, therefore they should be preserved when refreshing GRSF, because the generation of new ones, would break the former links.

4.2 Refreshing Workflow

Figure 5 shows the GRSF refreshing workflow. Similarly to the construction process, which has been described in Tzitzikas et al. (2019), and is also shown in the activity diagram in Figure 3, everything starts by harvesting and transforming data from the original data sources. Specifically, they are downloaded and transformed as ontology-based instances of the extended top level ontology MarineTLO (Tzitzikas et al. 2016). These instances are then ingested into a triplestore for constructing the new GRSF records (GRSF KB - V2). These activities are carried out by reusing or adapting existing software modules like MatWare (Tzitzikas et al. 2014), and X3ML Framework (Marketakis et al. 2017), and using software that has been implemented for the problem at hand, i.e. `grsf-services` and `grsf-publisher` [m].

Algorithm 1 shows how the VRE catalogue URLs and the manually-edited information are preserved across the two versions of GRSF KBs. More specifically, $GRSF\_new$ which is the new version and $GRSF\_pre$ which is the previous one. It traverses through the records in the new version of GRSF KB and finds their older instances in the previous version by inspecting their $r.sourceID$ *(lines 1-3)*. If the record is of type *Stock* then it replicates the catalogue URLs (i.e. $r.catalogID$), as well as all the editable information that have been updated by GRSF administrators in $GRSF\_pre$ (denoted by $r.info$). $r.info$ embodies all the fields of a record that can be edited by administrators *(lines 4-6)*. Since these updates are kept in GRSF KB and are not reflected in the original sources, their preservation in GRSF is crucial. Furthermore, administrators have the ability to propose merging multiple records into a new one, bypassing therefore the default merging algorithm that is being used.

For the case of records of type *fishery* an alternative approach is being followed, because of the dissection process carried out when constructing GRSF *fishery* records. Unlike *stock* records, if a source *fishery* record has multiple values over some specific fields, then they are dissected to construct GRSF fishery records, as depicted in Figure 2. As a result, because of the dissection process, the original URL of the fishery record is not enough for identifying the referring GRSF fishery record. For this reason, we are using the semantic ID as well. As described in §2.1, the semantic ID of fishery records is the concatenation of the values of five particular fields. Therefore, we compare those and identify a positive match if $r\_new.semanticID$ is an expansion of $r\_pre.semanticID$ *(lines 11-13)*. Of course the dissection of GRSF Fishery records is not always triggered. More specifically, it is applied only when there are multiple values for at least one of the fields: species, fishing gear and flag state. Clearly if a fishery record does not contain multiple values for any of the aforementioned fields, then the dissection process will create a single GRSF fishery record. As a result, for those cases, it does not make any sense to invoke the partial matching algorithm *(lines 7-10)*.

Algorithm 2 shows exactly how we implement the partial matching over the semantic IDs. An indicative example of such a partial match is given below, where the previous version of the semantic ID did not contain values for the last two fields. We should note here that

[m] `https://wiki.gcube-system.org/index.php?title=GCube_Data_Catalogue_for_GRSF`

Algorithm 1: Refreshing GRSF KB

Input: Collection $GRSF\_new$, Collection $GRSF\_pre$
Output: Collection $GRSF\_new$

```
1  forall r_new ∈ GRSF_new do
2      forall r_pre ∈ GRSF_pre do
3          if r_new.sourceID == r_pre.sourceID
4              if r_new.type == Stock
5                  r_new.catalogID = r_pre.catalogID
6                  r_new.info = r_pre.info
7              else if r_new.type == Fishery
8                  if r_new.is_dissected == false
9                      r_new.catalogID = r_pre.catalogID
10                     r_new.info = r_pre.info
11                 else if partialMatch(r_new.semanticID, r_pre.semanticID)
12                     r_new.catalogID = r_pre.catalogID
13                     r_new.info = r_pre.info
14 Return GRSF_new
```

this is usual, since as the data sources themselves evolve, missing information are added to them.

$r\_pre.semID$:

`asfis:GHL+rfb:NEAFC+auth:INT:NEAFC++`

$r\_new.semID$:

`asfis:GHL+rfb:NEAFC+auth:INT:NEAFC+iso3:GRL+isscfg:03.29`

The first step is to dissect the semantic ID into the concrete identifiers that are used for constructing it. The order of appearance of each identifier that is used for constructing the semantic ID is well-defined, which means that the identifier of the species will be the first part, the identifiers of the assessment and/or fishing areas will be the second, and so on. Then we compare the identifiers using different approaches. The proposed algorithm is the result of debating with data source maintainers, that know exactly which updates are carried out in their databases, and how such updates affect the record. More specifically:

• *species (lines 3-4):* GRSF fisheries contains a single species, so the semantic ID will contain a single species identifier. The identifiers that concerns species should be the same in both semantic IDs.

• *fishing areas (lines 5-6):* GRSF fisheries may contain multiple fishing areas, therefore the semantic ID will contain a concatenation of fishing areas IDs, using the symbol ';' as a separator. We decatenate those IDs, constructing therefore two sets of fishing areas IDs. If the two sets share at least on commmon ID (i.e. practically this means that the corresponding GRSF fishery records share at least a common area), then they are matched. This is carried out because updates in fishing areas are usually the result of improvements in the geographic accuracy of a fishery.

• *management authorities (lines 7-8):* GRSF fisheries may contain multiple management authorities, therefore the semantic ID will contain a concatenation of their IDs. We determine a matching based on the management authorities IDs, if: (a) the ID from the previous semantic ID is missing, or (b) there is at least one common ID in the two sets that are produced from the decatenation of the management authorities IDs. As regards the former we want to capture the cases where information for management authorities was missing but it has been added in newer versions, while for the latter we want to capture cases where information about management authorities has been fine-tuned while updating records in their original sources.

• *flag state & fishing gear (lines 9-12):* GRSF fisheries may contain a single flag state, or fishing gear because of the dissection process. As a result the semantic ID, will contain at most one ID of each type. We determine a matching based on the IDs for flag state if: (a) the ID from the previous semantic ID is missing , or (b) the IDs exists and are similar. We apply the same for fishing gear IDs.

Table 2 shows the results of the partial matching algorithm for some indicative semantic IDs. We should clarify here that we compare the concrete IDs lexicographically. This means that the aforementioned algorithm is not capable of determining a matching if different identifier types are used for the same resource. For example the species with scientific name Thunnus albacarres can be identified using the ASFIS code YFT (e.g. asfis:YFT), or the APHIAID 127027 (e.g. aphiaid:127027). Although, the partial matching algorithm does not capture such cases, it is not actually necessary to do so (at least for the purposes of GRSF). The reason for this, is because the workflow that is followed for constructing and refreshing GRSF and its semantic IDs, relies on the use of standard values for identifying resources (as described in §2.2) . This process ensures that the same ID from the same standard will be used in GRSF, no matter how a record is maintained in its original data source.

The activities carried out so far, resulted in the creation of a new version of the GSF KB. Now, we have to update the VRE catalogues. There are three sub-activities at this point: (a) updating the records that are

Algorithm 2: Partial matching of semantic IDs

Input: String *semanticID_pre*, *semanticId_new*
Output: boolean

```
semID_pre[] = semanticID_pre.decatenate('+')
semID_new[] = semanticID_new.decatenate('+')
if semID_pre[0]! = semID_new[0]
    return false
else if semID_pre[1].decatenate(';') ∩ semID_new[1].decatenate(';') == ∅
    return false
else if !semID_pre[2].empty() && semID_pre[2].decatenate(';') ∩ semID_new[2].decatenate(';') == ∅
    return false
else if !semID_pre[3].empty() && semID_pre[3]! = semID_new[3]
    return false
else if !semID_pre[4].empty() && semID_pre[4]! = semID_new[4]
    return false
return true
```

Semantic Identifiers	Res
`asfis:MAC+fao:27+authority:INT:EC+iso3:PRT+isscfg:03.29` `asfis:MAC+fao:27+authority:INT:EC+iso3:PRT+isscfg:03.29`	Yes
`asfis:CEX+fao:41.1.1;fao:41.2.2+authority:NAT:BRA+iso3:BRA+isscfg:03.19` `asfis:CEX+fao:41.1.1;fao:41.1.2+authority:NAT:BRA+iso3:BRA+isscfg:03.19`	Yes
`asfis:SZX+fao:71+authority:NAT:VNM+iso3:VNM+` `asfis:SZX+fao:71+authority:NAT:VNM+iso3:VNM+isscfg:03.19`	Yes
`asfis:ANE+fao:27+authority:NAT:PRT+iso3:PRT+isscfg:01.1` `asfis:ANE+fao:27+authority:INT:EC;authority:NAT:PRT+iso3:PRT+isscfg:01.1`	Yes
`asfis:HSO+fao:77+authority:NAT:PAN+iso3:PAN+isscfg:01.1` `asfis:HSO+fao:77+authority:NAT:PAN+iso3:COL+isscfg:01.1`	No
`asfis:SZX+fao:71+authority:NAT:VNM+isscfg:03.19` `asfis:SZX+fao:71+authority:NAT:VNM+iso3:VNM+`	No

Table 2 Results of the Semantic IDs partial matching algorithm

already published, (b) publishing new records that do not exist in the catalogues (c) remove or archive obsolete records.

The first group contains all the GRSF records, for which, we have identified their catalogue URLs, while the second one contains new records not yet assigned a catalogue URL. The former are updated (using their catalogue URLs), and the latter are published (a new catalogue URL is generated). The third group contains the obsolete records. The decision we have taken for obsolete records is to remove them from the catalogue, only if their status was not approved. The approved records are not removed from the catalogue with the rationale, that an approved record might have been disseminated publicly to external users or communities, so removing it would be an arbitrary decision. On the contrary, they are archived with a proper annotation message. We do not apply this for records under pending status; those records can be safely removed, since their status (pending) reveal that they have not been assessed by GRSF administrators.

5 Results and Evaluation

The refresh workflow that we propose meets all requirements described in §2.3. Obviously it tackles the refresh requirement *R1*. Most importantly, it preserves the work carried out by GRSF administrators, so as to maintain all of their inputs after refreshing and reconstructing GRSF. For example, updates in record names, traceability flags, connections, proposed merging, addition of narrative annotations, etc. (*req. R4, R5, R6*). In addition, the records that are obsolete are removed from GRSF KB and VRE catalogues (*req. R2*). Regarding obsolete records that were publicly available, they are properly archived. As a result, they are still publicly available, however their status, which is archived, reveals that they might not be valid any more. They are only kept in order to avoid creating broken URLs and as an historical evidence of their existence (*req. R3*).

From a technical perspective, the technical architecture of the refresh workflow relies on loosely-coupled technical components that are extensible and easy to maintain. Moreover, the entire process runs in a semi-automatic manner, which requires little human

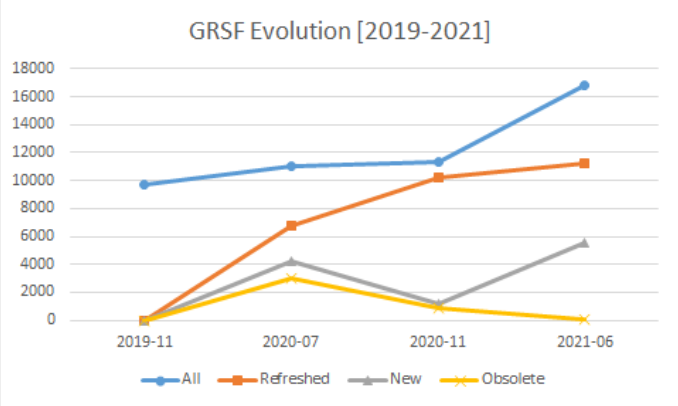

Figure 6 The overall statistics in terms of new, obsolete and refreshed records across 3 GRSF refreshes

intervention: the only step that human intervention is required is during the archival of obsolete records (e.g. for drafting a proper annotation message). This allows the entire process to be executed periodically.

As regards GRSF, and based on the contents it contains so far (i.e. information from three data sources; FIRMS, RAM, FishSource) the actual refreshing activity takes approximately 27 minutes to complete. Of course the aforementioned time includes the activites of exporting the URLs, the IDs, and the manually added information from the previous version of GRSF KB, applying the preserved infromation in the fresh contents, and re-constructing and adding the corresponding GRSF records in the new version of the GRSF KB. Moreover, the reported time does not include the time that is required for harvesting new data from the data sources, as well as the time for publishing the records in the catalogs of the GRSF VREs.

By the time of writing this paper (June 2021), we have carried out 3 refreshments of GRSF. Figure 6 reports for each version, (a) the total number of GRSF records, (b) the obsolete records, (c) the refreshed GRSF records, and (d) the new records that were found. It is evident that although the refreshments are carried out on a periodical basis (currently two refreshes per year), there is a significant number of obsolete and new records, demonstrating that the data sources that constitute GRSF constantly evolve, and our proposed refresh framework is capable enough of following this evolution. Table 6 shows the actual numbers (i.e. new, refreshed, obsolete and total records) for each GRSF version.

	Number of Records			
Release	**Refreshed**	**New**	**Total**	**Obsolete**
2019-11	-	-	-	9755
2020-07	6,757	4,251	11,008	2,998
2020-11	10,186	1,160	11,346	822
2011-06	11,284	5,531	16,815	62

Table 3 Refresh statistics

6 Applicability

In this section we discuss about the way updates in the semantic warehouse are preserved, and how some of them are propagated to the original sources (in §6.1), and a generalization of the evolution requirements (in §6.2).

6.1 Dealing with updates in the Semantic Warehouse

The core functionality of the refresh workflow that is described in this paper, is that it preserves the updates that have been carried out in the transformed data, after they have been semantically integrated. As we have already described, some of these updates contain information that is relevant for the integrated dataset (i.e. the GRSF dataset), such as the annotations made by GRSF administrators, the merging with other records, the semantic identifier. However, GRSF records might undergo with updates during their construction, and such updates are worth applying in their original data sources. As an example, consider the case where a GRSF record, is the result of the merging of two source records (from different data sources). Considering that they are merged, this means that they share the same information about marine species, however in the first source record, species is referred using its ASFIS code (e.g. ASFIS:YFT), while the second source record uses its common name in English language (e.g. Yellowfin tuna.). During the construction of GRSF records, information that contain standard codes are prioritized so the merged GRSF record will contain the ASFIS code for that species.

In this (not entirely artificial) example, the data source of the second legacy record is missing information about the standard code of the species with respect to ASFIS, and that would be a valuable addition for that source. In addition, compared to the GRSF only-related information (i.e. annotation messages, semantic identifiers, etc.), this type of information (i.e. the species update) is compliant with the data model and the contents of the original data sources. To this end, we could propagate those updates back in the original sources. So practically, this will enable a two-way exchange of information between the original data sources and the semantic warehouse; in one direction the data sources provide their contents to the semantic warehouse, and in the opposite direction the semantic warehouse provides the data sources with corrections and updated information.

Technically, this is a manual assessment carried out from GRSF Administrators after constructing or refreshing GRSF. More specifically, after constructing the refreshed version of the semantic warehouse we compare the GRSF records with their corresponding legacy records to spot such differences. The fields that will be used for comparison can be configured. For the case of GRSF, we apply the comparison over the same fields that are used for constructing the semantic

identifier of the records; species and water area for stock records and species, water areas, management authorities, fishing gears, and flag states for fishery records. So at the end of the refresh process, apart form the refresh report, that provides information about the obsolete, new and refreshed records, it will generate a set of update suggestions for each data source, by accumulating all the differences that are found for each data source.

Moreover, this bidirectional exchange of information between the data sources and the semantic warehouse, can also enhance the data sources themselves. For example, the overall aim of GRSF is to be a registry that can be used as a global reference for stocks and fisheries. This is the main requirement for assigning and preserving during the refresh the unique identifiers of GRSF records (i.e. UUIDs). The maintainers of the data sources contributing to GRSF, can store those identifiers in their databases (e.g. the latest version of the RAM database contains all the associated GRSF UUIDs) as a reference that the information they hold was used for constructing a GRSF record with that UUID. So even if someone is not aware of GRSF and works solely with any of the original data sources, he/she can find more details and connected information about it (e.g. if they are merged with records from other data sources), using such references.

6.2 A generalization of the problem

So far, we have described the requirements and the proposed refresh workflow for the case of GRSF. Despite the fact that the evolution requirements described in §2.3, are described in GRSF-related terms and seem to be applicable only for that case, in this section we provide a more generic list, that is applicable for other semantic warehouses as well, either domain-specific or generic ones. For this reason, below we provide a list of generic requirements, stemming from the GRSF-related ones, in order to refresh the contents of a semantic warehouse opposed to constructing it from scratch.

- (GR1): *Refresh* the contents of the semantic warehouse, with up-to-date information and resources from the underlying data sources. The contents of the data sources that are used for constructing a semantic warehouse continue to evolve themselves. It is evident that the semantic warehouse should be refreshed periodically to guarantee that it contains all the updates and fixes stemming from the contents of the original sources.
- (GR2): *Remove obsolete information* in a consistent way based on the updated information of the data sources. This means that the warehouse should ensure that it does not contain any resources that no longer exist in the data sources, or if they cannot be removed keep them in the semantic warehouse with a proper annotation. As regards the latter, we refer to the case of resource URIs/URLs from the semantic warehouse that has been exploited from third-party services and applications, and their removal will create non-resolvable resource URIs/URLs. The handling of the obsolete information should ensure the consistency of the resources of the semantic warehouse.
- (GR3): *Preserve the generated URIs* of the resources in the semantic warehouse. This is a strong requirement especially if the contents of the warehouse are resolvable URIs (or URLs) and are further disseminated or used from other services and applications. It is therefore required to avoid the generation of new URIs for resources that already exist in the semantic warehouse. For those, the evolution workflow should ensure that the previous URI will be reused, or it will reconstruct the same URI. If this cannot happen, due to technical limitation or any other problems, the semantic warehouse should provide a mechanism for resolving older URIs (e.g. by maintaining mappings between URIs).
- (GR4): *Maintain the updates* that have been carried out only in the semantic warehouse. These updates could be changes because of errors existing in the original data sources (e.g. misspelling of words), or changes in the data for offering an homogenized view (e.g. change the order of appearance of person names using as first term the surname). Despite, the fact that some of those changes could be propagated in the original sources, for example for the case of errors, the evolution framework should preserve the updated information, with the rationale that those updates are not reflected in the original data sources. To preserve those, the evolution framework should be informed about the fields that contain updated information so that they will not be disregarded.
- (GR5): *Preserve all the new information* that have been added in the semantic warehouse and does not exist in any of the underlying sources. These additions could be annotations of existing resources, supplementary information, etc. Usually they are associated with resources in the semantic warehouse, which means that apart from the information per se, the evolution framework should preserve their connection with the appropriate resources as well. As regards the latter, the preservation of the generated URIs (i.e. requirement *GR3*) will facilitate this.
- (GR6): *Preserve merges and combinations* of information that are used for deriving particular resources or new knowledge in the semantic warehouse. Usually, the resources of a semantic warehouse follow a reconciliation approach in order to find similarities between them, and then are merged to compile a new resource (e.g. construction of a resource about El Greco by merging information coming from different data sources), or combining information from different sources to deduce new knowledge (e.g. combining information about genes to define similarity metrics and derive new information about the connection between genes and diseases).

Figure 7 illustrates how the generic requirements of the refresh workflow are handled when datasets evolve. In particular, at the upper left part of the figure, a

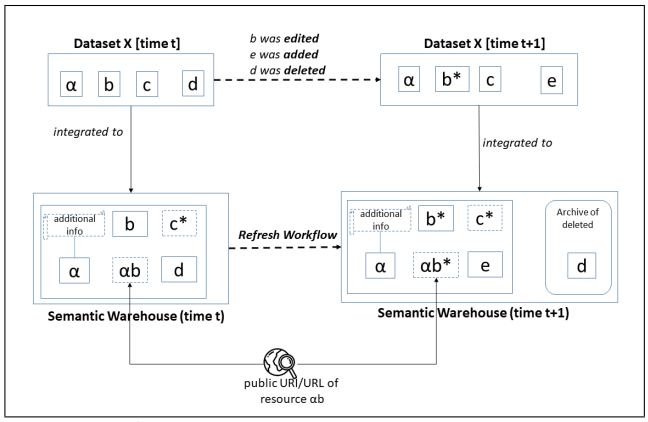

Figure 7 An overview of the refresh workflow that preserves edits and addition at the semantic warehouse

dataset is shown containing several resources (e.g. a, b, c, and d) and at the upper right part the newer version of the dataset that consists of updated resources (e.g. b^*), deleted resources (e.g. d) and new resources (e.g. e). The bottom left part shows these resources after integrating them into a semantic warehouse. Apart from integrating them, the reader can notice that resource a is enhanced with additional information, a new resource ab has been created using information from resources a and b, and resource c^* has updated information at the warehouse level. The bottom right part shows how the semantic warehouse has been refreshed with respect to the generic requirements. More specifically, all the contents of the semantic warehouse have been updated (*GR1*), the obsolete resource d is properly archived (*GR2*), the URIs of the resources at warehouse level has been preserved (*GR3*), the updated resource c^* has maintained its updates (*GR4*), the additional information added in resource a are preserved (*GR5*), and the resource ab that is the result of the combination of other resources is properly preserved (*GR6*) The aforementioned generic requirements (GR1-GR6) for supporting the evolution of a semantic warehouse, are stemming from our experience with GRSF. The provided list, aims to support the reader, in capturing the requirements for his/her own scenario/domain and configure the algorithms and the refresh workflow according to his/her needs.

7 Concluding Remarks

In this paper we have focused on the evolution requirements of a semantic warehouse about fish stocks and fisheries. We analyzed the associated requirements and then we described a process for tackling them. A distinctive characteristic of the approach is that it preserves all the manually added/edited information (at warehouse level), while at the same time it maintains the automation of the refresh process. In addition, we described how the original sources can benefit from such updates at the warehouse level, by spotting errors or anomalies with the data. The proposed solution is currently applied in the context of the ongoing EU project BlueCloud, where the aim for GRSF per se, is to to continue its evolution, as well as its expansion with more data sources and concepts (e.g. fish food and nutrition information). Despite the fact, that we focused on the case of stocks and fisheries, the same approach can be useful also in other domains where edits are required and allowed at the level of aggregated/integrated data.

One direction that is worth further work and research is to investigate how the semantically integrated, curated and manually enhanced data that exist in the semantic warehouse, can be used as a source for creating a new releases of the original data sources in an automatic manner. This includes the generation of the reverse schema mappings (e.g. map classes and properties of semantic warehouse ontology to resources from the data source schemata). The apparent benefit will be that each data source will have a new improved version (e.g. with standards values) right after the construction of the semantic warehouse.

Acknowledgement

This work has received funding from the European Union's Horizon 2020 innovation action BlueCloud (Grant agreement No 862409).

References

Assante, M., Candela, L., Castelli, D., Cirillo, R., Coro, G., Frosini, L., Lelii, L., Mangiacrapa, F., Pagano, P., Panichi, G. et al. (2019), 'Enacting open science by d4science', *Future Generation Computer Systems* **101**, 555–563.

Dividino, R. Q., Gottron, T., Scherp, A. & Gröner, G. (2014), From changes to dynamics: Dynamics analysis of linked open data sources., *in* 'Proceedings of PROFILES@ESWC', CEUR-WS.org.

Dos Reis, J. C., Pruski, C., Da Silveira, M. & Reynaud-Delaître, C. (2015), 'Dykosmap: A framework for mapping adaptation between biomedical knowledge organization systems', *Journal of biomedical informatics* **55**, 153–173.

Flouris, G., Manakanatas, D., Kondylakis, H., Plexousakis, D. & Antoniou, G. (2008), 'Ontology change: Classification and survey', *The Knowledge Engineering Review* **23**(2), 117–152.

Hyvönen, E., Heino, E., Leskinen, P., Ikkala, E., Koho, M., Tamper, M., Tuominen, J. & Mäkelä, E. (2016), Warsampo data service and semantic portal for publishing linked open data about the second world war history, *in* 'European Semantic Web Conference', Springer, pp. 758–773.

Jaradeh, M. Y., Oelen, A., Farfar, K. E., Prinz, M., D'Souza, J., Kismihók, G., Stocker, M. & Auer, S. (2019), Open research knowledge graph: Next generation infrastructure for semantic scholarly knowledge, *in* 'Proceedings of the 10th International Conference on Knowledge Capture', pp. 243–246.

Käfer, T., Abdelrahman, A., Umbrich, J., O' Byrne, P. & Hogan, A. (2013), Observing linked data dynamics, *in* 'Extended Semantic Web Conference', Springer, pp. 213–227.

Kondylakis, H. & Plexousakis, D. (2013), 'Ontology evolution without tears', *Web Semantics: Science, Services and Agents on the World Wide Web* **19**, 42–58.

Lantzaki, C., Papadakos, P., Analyti, A. & Tzitzikas, Y. (2017), 'Radius-aware approximate blank node matching using signatures', *Knowledge and Information Systems* **50**(2), 505–542.

Marketakis, Y., Minadakis, N., Kondylakis, H., Konsolaki, K., Samaritakis, G., Theodoridou, M., Flouris, G. & Doerr, M. (2017), 'X3ml mapping framework for information integration in cultural heritage and beyond', *International Journal on Digital Libraries* **18**(4), 301–319.

Marketakis, Y., Tzitzikas, Y., Gentile, A., van Niekerk, B. & Taconet, M. (2021), 'On the evolution of semantic warehouses: The case of global record of stocks and fisheries', *Metadata and Semantic Research* **1355**, 269.

Mountantonakis, M., Minadakis, N., Marketakis, Y., Fafalios, P. & Tzitzikas, Y. (2016), 'Quantifying the connectivity of a semantic warehouse and understanding its evolution over time', *IJSWIS* **12**(3), 27–78.

Mountantonakis, M. & Tzitzikas, Y. (2019), 'Large-scale Semantic Integration of Linked Data: A Survey', *ACM Computing Surveys (CSUR)* **52**(5), 103.

R. Gazzotti, F. Michel, F. G. (2020), 'Cord-19 named entities knowledge graph (cord19-nekg)'.

Regino, A. G., dos Reis, J. C., Bonacin, R., Morshed, A. & Sellis, T. (2021), 'Link maintenance for integrity in linked open data evolution: Literature survey and open challenges', *Semantic Web* (Preprint), 1–25.

Reis, R. B., Morshed, A. & Sellis, T. (2019), 'Understanding link changes in lod via the evolution of life science datasets'.

Roussakis, Y., Chrysakis, I., Stefanidis, K., Flouris, G. & Stavrakas, Y. (2015), A flexible framework for understanding the dynamics of evolving rdf datasets, *in* 'International Semantic Web Conference', Springer, pp. 495–512.

Tzitzikas, Y., Allocca, C., Bekiari, C., Marketakis, Y., Fafalios, P., Doerr, M., Minadakis, N., Patkos, T. & Candela, L. (2016), 'Unifying heterogeneous and distributed information about marine species through the top level ontology marinetlo', *Program* **50**(1), 16–40.

Tzitzikas, Y., Kampouraki, M. & Analyti, A. (2013), 'Curating the Specificity of Ontological Descriptions under Ontology Evolution', *Journal on Data Semantics* pp. 1–32.

Tzitzikas, Y., Marketakis, Y., Minadakis, N., Mountantonakis, M., Candela, L., Mangiacrapa, F. et al. (2019), Methods and tools for supporting the integration of stocks and fisheries, *in* 'Chapter in Information and Communication Technologies in Modern Agricultural Development, Springer, 2019.', Springer.

Tzitzikas, Y., Minadakis, N., Marketakis, Y., Fafalios, P., Allocca, C., Mountantonakis, M. & Zidianaki, I. (2014), Matware: Constructing and exploiting domain specific warehouses by aggregating semantic data, *in* 'ESWC', Springer, pp. 721–736.

Vandenbussche, P.-Y., Umbrich, J., Matteis, L., Hogan, A. & Buil-Aranda, C. (2017), 'Sparqles: Monitoring public sparql endpoints', *Semantic web* **8**(6), 1049–1065.

Wang, L. L., Lo, K., Chandrasekhar, Y., Reas, R., Yang, J., Eide, D., Funk, K., Kinney, R., Liu, Z., Merrill, W. et al. (2020), 'Cord-19: The covid-19 open research dataset', *ArXiv* .

Wishart, D. S., Feunang, Y. D., Guo, A. C., Lo, E. J., Marcu, A., Grant, J. R., Sajed, T., Johnson, D., Li, C., Sayeeda, Z. et al. (2018), 'Drugbank 5.0: a major update to the drugbank database for 2018', *Nucleic acids research* **46**(D1), D1074–D1082.

Yumusak, S., Dogdu, E., Kodaz, H., Kamilaris, A. & Vandenbussche, P. (2017), 'Spend: Linked data sparql endpoints discovery using search engines', *IEICE TRANSACTIONS on Information and Systems* **100**(4), 758–767.

Zeginis, D., Tzitzikas, Y. & Christophides, V. (2011), 'On computing deltas of rdf/s knowledge bases', *ACM Transactions on the Web (TWEB)* **5**(3), 1–36.