


Preprint of:

Y. Marketakis, Y. Tzitzikas, A. Gentile, B. van Niekerk and M. Taconet,
On the Evolution of Semantic Warehouses: The Case of Global Record of Stocks and Fisheries,
14th Metadata and Semantics Research Conference (MTSR 2020), Nov 2020.

On the Evolution of Semantic Warehouses: The Case of Global Record of Stocks and Fisheries

Yannis Marketakis ¹[0000-0002-0417-2526], Yannis Tzitzikas^{1,2}[0000-0001-8847-2130], Aureliano Gentile³[0000-0002-6542-132x], Bracken van Niekerk³[0000-0001-8537-3305], and Marc Taconet³[0000-0002-3103-6204]

¹ Institute of Computer Science, FORTH-ICS, Heraklion, Greece

² Computer Science Department, University of Crete, Heraklion, Greece
{marketak,tzitzik}@ics.forth.gr

³ Food and Agriculture Organization of the United Nations, Rome Italy
{aureliano.gentile,bracken.vanniekerk,marc.taconet}@fao.org

Abstract. Semantic Warehouses integrate data from various sources for offering a unified view of the data and enabling the answering of queries which cannot be answered by the individual sources. However, such semantic warehouses have to be refreshed periodically as the underlying datasets change. This is a challenging requirement, not only because the mappings and transformations that were used for constructing the semantic warehouse can be invalidated, but also because additional information (not existing in the initial datasets) may have been added in the semantic warehouse, and such information needs to be preserved after every reconstruction. In this paper we focus on this particular problem in a real setting: the Global Record of Stocks and Fisheries, a semantic warehouse that integrates data about stocks and fisheries from various information systems. We propose and detail a process that can tackle these requirements and we report our experiences from implementing it.

1 Introduction

The Web of Data contains thousands of RDF datasets available online (see [12] for a recent survey), including cross-domain Knowledge Bases (e.g., DBpedia and Wikidata), domain specific repositories (e.g., WarSampo [5], DrugBank [21], ORKG [6], life science related datasets [14] and recently COVID-19 related datasets [8,13]), as well as Markup data through `schema.org`. One important category of domain specific semantic repositories, are the semantic warehouses, those 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 because the underlying datasets change since they are managed by different stakeholders and operating information systems. This is a challenging requirement, not only because the mappings and transformations that were used for constructing the semantic warehouse can be invalidated, but

also because additional information, that does not exist in the initial datasets, may have been added in the semantic warehouse, and such information needs to be preserved after every reconstruction. 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 (for short GRSF) [19], a semantic warehouse that integrates data about stocks and fisheries from various information systems. In brief, GRSF is capable of hosting the corresponding information categorized into uniquely and globally identifiable records. Instead of creating yet another registry, GRSF aims at producing its records by using existing data. This approach 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.

In this paper, we focus on the evolution of this domain-specific semantic warehouse. 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 from 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. In a nutshell, the key contributions of this paper are: (a) the analysis of the requirements for preserving updates in aggregated records that are not reflected in the original data, (b) the identification of aggregated records in different versions of a semantic warehouse, (c) the definition of a process for managing the evolution while preserving updates in the aggregated records.

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, and Section 5 reports our experience on the implementation. Finally Section 6 concludes the paper and elaborates with future work and research.

2 Context

Here we provide background information about the domain-specific warehouse GRSF (in §2.1) and then discuss the evolution-related requirements (in §2.2).

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 BlueBRIDGE⁴. It integrates data from three different data sources, owned by different stakeholders, in a knowledge base (the GRSF KB), and then exposes them through a catalogue of a Virtual Research Environment (VRE), operated on top of D4Science infrastructure[1]. These data sources are: (a) Fisheries and Resources Monitoring System (FIRMS)⁵, (b) RAM Legacy Stock Assessment database⁶, and (c) FishSource⁷. They contain complementary information (both conceptually and geographically). Specifically, FIRMS is mostly reporting about stocks and fisheries at regional level, while RAM is reporting stocks at national or subnational level, and FishSource is more focused on the fishing activities. All of them contribute to the overall aim to build a comprehensive and transparent global reference set of stocks and fisheries records that will boost regional and global stocks and fisheries status and trend monitoring as well as responsible consumer practices. GRSF continues its evolution and expansion in the context of the ongoing H2020 EU Project BlueCloud⁸.

GRSF intends to collate information in terms of *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 [19]. 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 semantic warehouse 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 and they become available to external users. (c) *External users* for querying and browsing it. To ease understanding, Table 1 provides some background information about the terminologies of GRSF that are used in the sequel.

⁴ BlueBRIDGE (H2020-BG-2019-1), GA no 675680

⁵ <http://firms.fao.org/firms/en>

⁶ <https://www.ramlegacy.org/>

⁷ <https://www.fishsource.org/>

⁸ BlueCloud (H2020-EU.3.2.5.1), GA no: 862409

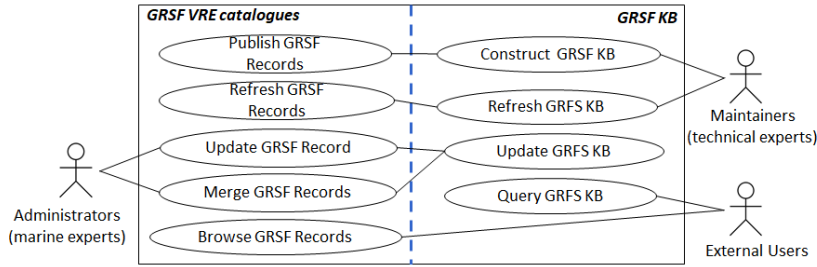


Fig. 1. Use Case Diagram describing the GRSF Ecosystem

Term	Description
Source Record	A record that has been derived by transforming its original contents, with respect to a core ontology, specifically MarineTLO [17]. 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 (e.g. where possible, instead of a species common name use the FAO ASFIS classification), generate new attributes (e.g. semantic ID), flags, citations, etc.
Semantic ID	They are <i>identifiers</i> 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+IS03:CAN).
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 4.
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. An example is shown in Figure 4.
Approved Record	After their construction GRSF records, appear with status <i>pending</i> . Once they are assessed from GRSF administrators, they can be approved (if they are valid) and as a result their status is changed to <i>approved</i> . Approved records are then made publicly available.

Table 1. Explanation of the Terminology in GRSF

Figure 2 shows the different activities that are carried out. Initially, information from the data sources are transformed and 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 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.

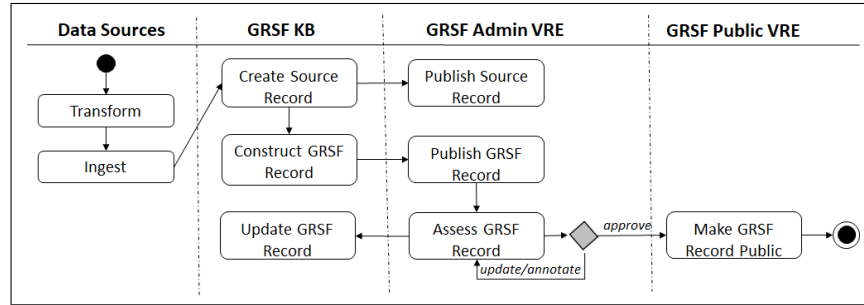


Fig. 2. The process of constructing, publishing and assessing GRSF records

2.2 Evolution Requirements

The following list provides the key requirements for refreshing GRSF:

- (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 administrators 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-producible 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 access in general. A survey for ontology evolution is given in [4]. 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 [9] where query re-writing methods are proposed. The losses of specificity of ontology-based descriptions, when such descriptions are migrated to newer versions of the ontology has been studied in [18]. Finally, there are various methods that focus on monitoring the “health” of various RDF-based systems, e.g. [11] focuses on the connectivity monitoring in the context of a semantic warehouse over time, [7] focuses on monitoring Linked Data over a specific period of time, [2] focuses on measuring the dynamics of a specific RDF dataset, and [15] proposes a framework that identifies, analyses and understands such dynamics. *SPARQLES* [20] and *SpEnD* [22] focus on the monitoring of public SPARQL endpoints, *DyKOSMap* framework [3] adapts the mappings of Knowledge Organization Systems, as the data are modified over time.

[14] 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). 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.

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 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 3 shows the GRSF refreshing workflow. Similarly to the construction process, which has been described in [19], and is also shown in the activity diagram in Figure 2, 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 [17]. 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 [16], and X3ML Framework [10], and using software that has been implemented for the problem at hand, i.e. `grsf-services` and `grsf-publisher`⁹.

⁹ https://wiki.gcube-system.org/index.php?title=GCube_Data_Catalogue_for_GRSF

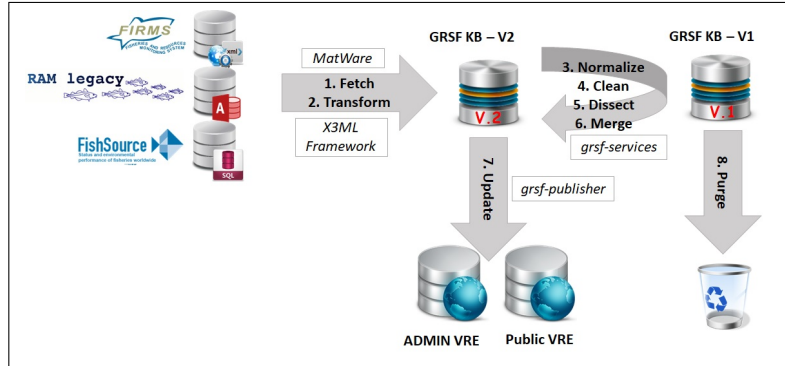


Fig. 3. The workflow for refreshing GRSF, while preserving particular information from the previous version

Algorithm 1: Refreshing GRSF KB

Input: Collection $GRSF_new$, Collection $GRSF_pre$

Output: Collection $GRSF_new$

```

1 forall  $r\_new \in GRSF\_new$  do
2   forall  $r\_pre \in GRSF\_pre$  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  $partialMatch(r\_new.semanticID, r\_pre.semanticID)$ 
9            $r\_new.catalogID = r\_pre.catalogID$ 
10           $r\_new.info = r\_pre.info$ 
11 Return  $GRSF\_new$ 

```

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 $r.sourceID$. 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. 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 4. The fields considered for the dissection process are species, fishing gears and flags states. Considering that the source fishery record example contains two different species, the dissection process produces two distinct GRSF fishery records.

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

r_pre.semanticID: asfis:GHL+rfb:NEAFC+auth:INT:NEAFC++

r_new.semanticID: asfis:GHL+rfb:NEAFC+auth:INT:NEAFC+iso3:GRL+isscfg:03.29

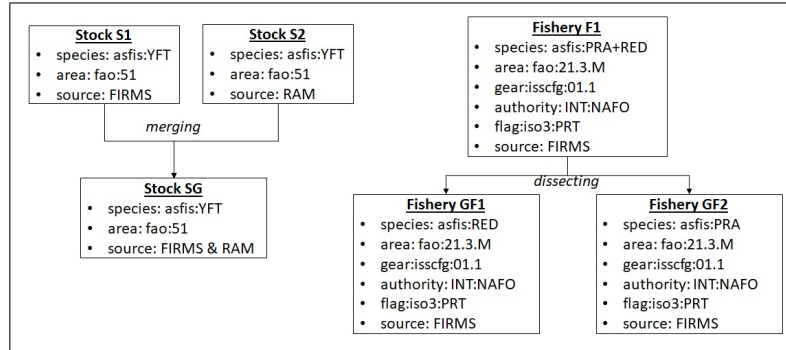


Fig. 4. Merging multiple stock records in a single GRSF stock record (left part) and dissecting a single fishery record in multiple GRSF fishery records (right part)

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 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 §2.2. 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 re-constructing 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 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.

Table 2 shows some statistics about the refresh. The original version was constructed on December 2018, and the refresh was carried out on July 2020. Figure 5 shows the time that is needed for each step of the refresh workflow. The most time-consuming step is the last one (i.e. Publish / Update) that publishes or (re-publishes) records in VRE catalogues because records are published in sequential manner, through the a set of VRE publishing services that unavoidably perform several validity checks, and each record takes around 2 seconds to be published. It is worth mentioning however, that when publishing/updating takes place, GRSF KB has already been refreshed, and this (last step of the entire process) just makes the records visible in GRSF catalogues. Another remark is that the actual refreshing activities that are part of the Refresh Source, Construct GRSF and Delete Archive steps, are performed rather quickly. Overall, the refresh workflow has a similar efficiency with the GRSF construction process.

6 Concluding Remarks

We have focused on the evolution requirements of a semantic warehouse about fish stocks and fisheries. We analyzed the associated requirements and then we

Source	Number of Records		
	Refreshed	New	Obsolete
FIRMS	928	122	5
RAM	1,290	84	1
FishSource	4,094	975	117
GRSF	6,690	7,318	2,998

Table 2. Refresh statistics

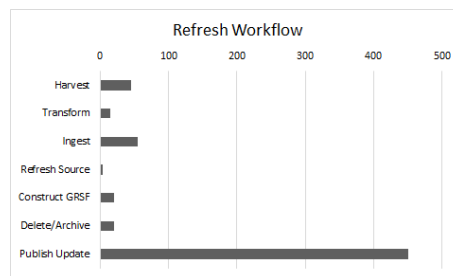


Fig. 5. Refresh Time (in minutes)

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. The proposed solution is currently applied in the context of the ongoing EU Project BlueCloud, where the aim for GRSF per se, is 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 allowed at the level of aggregates/integrated data.

Issues that are worth further work and research include: the partial refreshing of the semantic warehouse, which would be useful if there are data sources that are more frequently updated compared to others, the addition of generated information from the semantic warehouse (i.e. unique identifiers) back to the original sources in order to support the refreshing workflow and enforce their preservation, the automatic identification of the existence of updates in the underlying source which would trigger the refreshing workflow as well as the estimation of the refreshing period based on the update frequency of a data source that would enable the fully automatic trigger and execution of the refresh workflow.

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

References

1. Assante, M., Candela, L., Castelli, D., Cirillo, R., Coro, G., Frosini, L., Lelii, L., Mangiacrapa, F., Pagano, P., Panichi, G., et al.: Enacting open science by d4science. *Future Generation Computer Systems* **101**, 555–563 (2019)
2. Dividino, R.Q., Gottron, T., Scherp, A., Gröner, G.: From changes to dynamics: Dynamics analysis of linked open data sources. In: *Proceedings of PROFILES@ESWC*. CEUR-WS.org (2014)
3. Dos Reis, J.C., Pruski, C., Da Silveira, M., Reynaud-Delaître, C.: Dykosmap: A framework for mapping adaptation between biomedical knowledge organization systems. *Journal of biomedical informatics* **55**, 153–173 (2015)
4. Flouris, G., Manakanatas, D., Kondylakis, H., Plexousakis, D., Antoniou, G.: Ontology change: Classification and survey. *The Knowledge Engineering Review* **23**(2), 117–152 (2008)
5. Hyvönen, E., Heino, E., Leskinen, P., Ikkala, E., Koho, M., Tamper, M., Tuominen, J., Mäkelä, E.: Warsampo data service and semantic portal for publishing

- linked open data about the second world war history. In: European Semantic Web Conference. pp. 758–773. Springer (2016)
6. Jaradeh, M.Y., Oelen, A., Farfar, K.E., Prinz, M., D’Souza, J., Kismihók, G., Stocker, M., Auer, S.: Open research knowledge graph: Next generation infrastructure for semantic scholarly knowledge. In: Proceedings of the 10th International Conference on Knowledge Capture. pp. 243–246 (2019)
 7. Käfer, T., Abdelrahman, A., Umbrich, J., O’ Byrne, P., Hogan, A.: Observing linked data dynamics. In: Extended Semantic Web Conference. pp. 213–227. Springer (2013)
 8. Kohlmeier, S., Lo, K., Wang, L.L., Yang, J.: Covid-19 open research dataset (cord-19) (Mar 2020). <https://doi.org/10.5281/zenodo.3813567>, <https://doi.org/10.5281/zenodo.3813567>
 9. Kondylakis, H., Plexousakis, D.: Ontology evolution without tears. *Web Semantics: Science, Services and Agents on the World Wide Web* **19**, 42–58 (2013)
 10. Marketakis, Y., Minadakis, N., Kondylakis, H., Konsolaki, K., Samaritakis, G., Theodoridou, M., Flouris, G., Doerr, M.: X3ml mapping framework for information integration in cultural heritage and beyond. *International Journal on Digital Libraries* **18**(4), 301–319 (2017)
 11. Mountantonakis, M., Minadakis, N., Marketakis, Y., Fafalios, P., Tzitzikas, Y.: Quantifying the connectivity of a semantic warehouse and understanding its evolution over time. *IJSWIS* **12**(3), 27–78 (2016)
 12. Mountantonakis, M., Tzitzikas, Y.: Large-scale Semantic Integration of Linked Data: A Survey. *ACM Computing Surveys (CSUR)* **52**(5), 103 (2019)
 13. R. Gazzotti, F. Michel, F.G.: Cord-19 named entities knowledge graph (cord19-nekg) (2020)
 14. Reis, R.B., Morshed, A., Sellis, T.: Understanding link changes in lod via the evolution of life science datasets (2019)
 15. Roussakis, Y., Chrysakis, I., Stefanidis, K., Flouris, G., Stavrakas, Y.: A flexible framework for understanding the dynamics of evolving rdf datasets. In: International Semantic Web Conference. pp. 495–512. Springer (2015)
 16. Tzitzikas, Y., Minadakis, N., Marketakis, Y., Fafalios, P., Allocca, C., Mountantonakis, M., Zidianaki, I.: Matware: Constructing and exploiting domain specific warehouses by aggregating semantic data. In: ESWC. pp. 721–736. Springer (2014)
 17. Tzitzikas, Y., Allocca, C., Bekiari, C., Marketakis, Y., Fafalios, P., Doerr, M., Minadakis, N., Patkos, T., Candela, L.: Unifying heterogeneous and distributed information about marine species through the top level ontology marinetlo. *Program* **50**(1), 16–40 (2016)
 18. Tzitzikas, Y., Kampouraki, M., Analyti, A.: Curating the Specificity of Ontological Descriptions under Ontology Evolution. *Journal on Data Semantics* pp. 1–32 (2013)
 19. Tzitzikas, Y., Marketakis, Y., Minadakis, N., Mountantonakis, M., Candela, L., Mangiacrapa, F., et al.: 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 (2019)
 20. Vandenbussche, P.Y., Umbrich, J., Matteis, L., Hogan, A., Buil-Aranda, C.: Sparqls: Monitoring public sparql endpoints. *Semantic web* **8**(6), 1049–1065 (2017)
 21. 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.: Drugbank 5.0: a major update to the drugbank database for 2018. *Nucleic acids research* **46**(D1), D1074–D1082 (2018)
 22. Yumusak, S., Dogdu, E., Kodaz, H., Kamilaris, A., Vandenbussche, P.: Spend: Linked data sparql endpoints discovery using search engines. *IEICE TRANSACTIONS on Information and Systems* **100**(4), 758–767 (2017)