ever-est

Design, Implementation and Deployment of Research Objects Components for Earth Science Phase 1

Workpackage	**4**	**Research Objects in Earth Science**
Task (s)	4.3	Research Objects search and recommendation in Earth Science communities
	4.4	Preservation and curation of Research Objects in Earth Science: RO life cycle and quality management
	4.5	Social aspects of Research Objects in Earth Science: traceability, citation, trust, credit and attribution
Author (s)	Andrés García-Silva	ESI
	Jose Manuel Gomez-Perez	ESI
	Raul Palma	PSNC
Reviewer (s)	Fulvio Marelli	Terradue
	Rosemarie Leone	ESA
Approver (s)	Jose Manuel Gomez-Perez	ESI
	Cristiano Silvagni	ESA
Authorizer	Mirko Albani	ESA
Document Identifier	EVER-EST DEL WP4-D4.3	
Dissemination Level	Public	
Status	Draft to be approved by the EC	
Version	1.0	
Date of Issue	08/12/2016	

Abstract

One of the main motivations of research communities in EVER-EST to integrate the research object model in their scientific process is to enhance sharing and reuse of their scientific results. The research object model enables knowledge sharing and reuse; its foundations, including the common ontology and the data model, ensures data interoperability and reusability. Nevertheless, the model needs to be integrated in an ecosystem of tools that provides mechanism to search and explore research objects, assess their quality, and properly credit the authors when their research objects are reused by other scientists.

This deliverable describes the first release of the components that make up the ecosystem of tools complementing the research object model. These components include retrieval mechanisms to support search and exploration of research objects, checklists designed in cooperation with VRCs to assess the quality of research objects in Earth Science, digital object identifiers DOIs to uniquely and permanently identify research objects, and tools to keep track of citations of research objects in scholarly communications.

Document Log

Date	Author	Changes	Version	Status
7/10/2016	Andrés García-Silva	Table Of contents	0.1	Draft
21/10/2016	Andrés García-Silva	Search and recommender section	0.2	Draft
31/10/2016	Jose Manuel Gomez Perez	Revision of the search and recommender section	0.2.1	Draft
4/11/2016	Andrés García-Silva	Section about checklists for Earth Science	0.3	Draft
9/11/2016	Jose Manuel Gomez Perez	Revision of checklist section	0.3.1	Draft
10/11/2016	Jose Manuel Gomez Perez	Research object life cycle	0.4	Draft
11/11/2016	Andrés García-Silva	DOI and impact metrics section added	0.5	Draft
14/11/2016	Andrés García-Silva	Introduction, future work and section about research object components in EVER-EST architecture	0.6	Draft
14/11/2016	Raul Palma	Section about faceted search, keyword search and search API	0.7	Draft
15/11/2016	Andrés García-Silva	Generate first complete version of the deliverable	0.8	Draft
16/11/2016	Raul Palma	Modifications to the sections about Checklists, and Research object lifecycle	0.9	Draft
16/11/2016	Andrés García-Silva	Complete version ready for internal review	0.10	Draft
25/11/2016	Andrés García-Silva	Complete revision after internal review	0.11	Draft
30/11/2016	Andrés García-Silva	Conclusions and Future work added	1.0	Draft to be approved by the EC

Table of Contents

List of Figures

List of Tables

Definitions and Acronyms

Acronym	Description
CBOW	Continuous Bag of Words
DOI	Digital Object Identifiers
IDF	Inverse Document Frequency
IR	Information Retrieval
NLP	Natural Language Processing
RO	Research Object
RODL	Research Object Digital Library
ROEVO	Research Object Evolution Ontology
RO MODEL	Research Object Core Ontology
TF	Term Frequency
VRC	Virtual Research Community
VRE	Virtual Research Environment
W3C	World Wide Web Consortium
WFDESC	Workflow Description Ontology
WFPROV	Workflow Execution Provenance Ontology
WP	Work Package

Reference Documents

Document ID	Document Title
[1]	G. Salton, A. Wong, and C. S. Yang (1975), "A Vector Space Model for Automatic Indexing," *Communications of the ACM*, vol. 18, nr. 11, pages 613–620
[2]	Gerard Salton and Michael J. McGill. 1986. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA
[3]	Cai-Nicolas Ziegler, Sean M. McNee, Joseph A. Konstan, and Georg Lausen. 2005. Improving recommendation lists through topic diversification. In *Proceedings of the 14th international conference on World Wide Web* (WWW '05). ACM, New York, NY, USA, 22-32. DOI=http://dx.doi.org/10.1145/1060745.1060754
[4]	Peter D. Mikolov. 2006. Similarity of semantic relations. Computational Linguistics, 32(3):379–416
[5]	Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. In Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 746–751, Atlanta, Georgia, June. Association for Computational Linguistics.

[6]	Omer Levy, Yoav Goldberg. 2014. Linguistic Regularities in Sparse and Explicit Word Representations. CoNLL 2014: 171-180
[7]	Rafael González-Cabero, Raúl Palma (PSNC), Jose Gómez-Pérez, Aleix Garrido. 2013. D3.2v2 Design, implementation and deployment of Workflow Integrity and Authenticity Maintenance components - Phase II, Wf4Ever project.
[8]	Esteban García-Cuesta, Graham Klyne, Aleix Garrido, Jose Manuel Gómez-Pérez, Jun Zhao. 2013. D4.2: Design, implementation and deployment of Workflow Integrity and Authenticity Maintenance components – Phase II
[9]	Dong, Hai, Farookh Khadeer Hussain, and Elizabeth Chang. "A survey in semantic search technologies." 2nd IEEE International Conference on Digital Ecosystems and Technologies. 2008.

1 Introduction

The requirement analysis presented in D4.1 regarding a VRE based on the research object model showed that VRC expectations were mainly focused on knowledge sharing and reuse, and new forms of scholarly communications beyond pdf articles as supporting tools of knowledge cross-fertilization between their members. Enabling sharing and reuse of scientific knowledge was a key design principle of the research object model and these features are supported by the use of a common vocabulary (the research object model) and a W3C standard data format (the resource description framework) for information interchange. However, the research object model by itself is not enough; it requires to be integrated in an ecosystem of tools that enables the functionalities that need to be provided to Earth Science communities around research objects. Such functionalities are top on top of the research object model and include: research object search and exploration, quality assessment, and tracking of citation information in scholarly communication. This deliverable describes the components constituting the ecosystem of tools that complement the research object model.

The retrieval component includes a search engine and a recommender system based on the Collaboration Spheres metaphor. The search engine is an ongoing development where two different approaches are being tested: one that leverage the explicit metadata of research objects and a second one tapping into the research object content. While in this stage initial versions of such components are presented, the goal is that for the next release of the components (deliverable D4.4) the two different approaches are combined in one search engine that benefits of metadata and content. For the search engine that indexes the research object content, semantic technologies are used to enable an intelligent processing of natural language. This semantic analysis produces as output the identification of the research object main concepts, domains of work, and named entities such as people, organization and places.

The recommender system is the complement to the search engine since it supports the exploration of the research object collection. In the recommender system, users do not have to define what they are looking for explicitly but they are required to provide contextual material to drive the explorative search process. In a VRE context, explorative search of research objects is fundamental, e.g. for validation and reuse. The user profile elicited in the requirement analysis stage of the project showed that VRC members are very skilled scientist in their domain of work although they cannot be considered experts in the IT domain. Therefore, an assisted tool such as the recommender system is very useful for them as it hides the complexity of the advanced search carried out underneath.

Another important aspect that influences knowledge reusability is the research object quality. It is expected that high quality research objects are more reused than low quality ones. Inspired in best practices from wet lab disciplines, checklists are the main tool introduced to assess research object quality. This deliverable describes the joint work with the VRCs to define the checklists that include the main features expected in high quality research objects in their domains of interest.

Finally, the last component that boosts research object sharing and reuse in the context of scholarly communications is related to the uptake of the digital object identifier DOI for research objects in Earth Science and the management of its lifecycle. A DOI is a persistent unique identifier that resolves a persistent network link to current information about that object. Benefits of using DOIs to identify research objects include: i) research objects enhanced findability, since they are automatically indexed externally by prominent scholarly communication sites (e.g., Datacite), ii) research objects can be properly cited and automatically receive credit when they are reused, and iii) the impact of specific research objects in the scientific communities can be measured by tracking citations to their DOIs across publications by other peers. This deliverable describes the agreement reached with Datacite through its node at the British library for DOI generation, the new research object life cycle where DOIs are a central concept for releasing research objects, and the software tools that have been put in place to harvest citation information.

1.1 Relation to other work packages

This deliverable has direct relation to deliverables D3.1 (Virtual Research Environment detailed definition of use cases), D5.4 (Technical Note on e-Research application services), and D5.8 (Technical Note on e-research application services, Final version) of work packages 3 and 5 respectively (see the links between the corresponding work packages in Figure 1). D3.1 describes the uses cases representing the different needs of the research communities that are part of the project; from these use cases, a set of research object-specific requirements were obtained and used as the basis to design and implement the tools described in this deliverable. Some of the tools presented in the deliverable have application programming interfaces that are described in D5.4., and must be integrated in the VRE. Therefore, the D4.3 influence the forthcoming D5.8 where these integration aspects should be addressed.

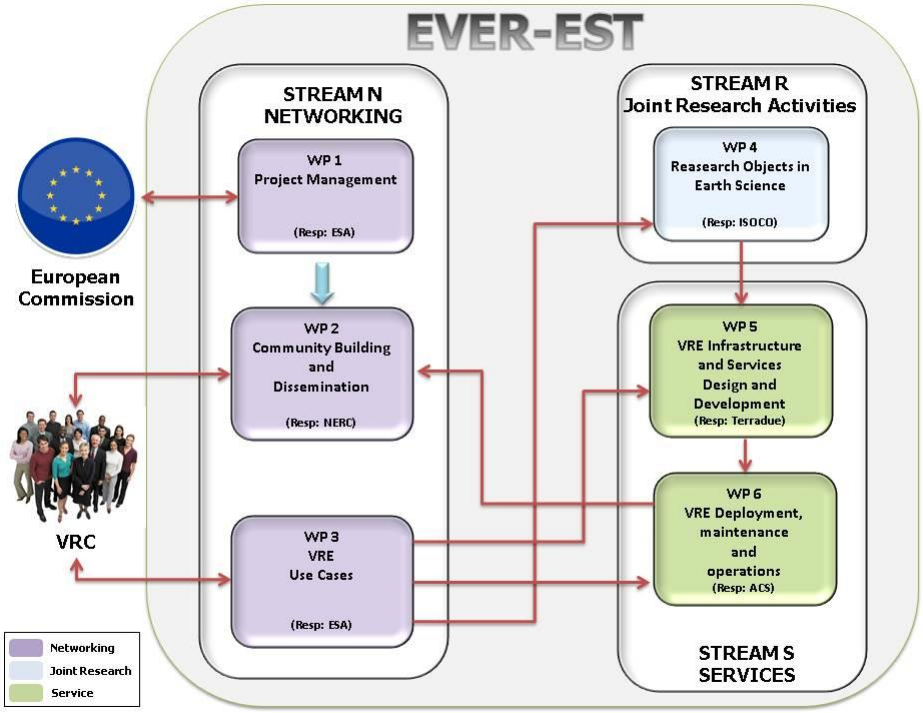

Figure 1. Work package dependencies

1.2 Compliance to the Smart Objectives and Key Performance Indicators

The contents of this deliverable are in line with Objective 3 of the project, i.e. the implementation and validation of the use of research objects in Earth Science. Of particular relevance are the second and the third SMART objective related with objective 3. In the SMART objective 2 the goal is to generate tools for research object preservation. This deliverable addresses two important issues regarding preservation: quality assessment of research objects using checklists, and the use of DOIs as persistent and unique identifiers for research objects. Designed checklists and DOI generation in the research object lifecycle are supported in ROHUB.

SM_OB#3.2	**Development of means and technological support for research object preservation, including decay diagnosis and prevention both at the method and implementation levels.**
Measured	Implementation of core research object management services, including storage,

by	retrieval, lifecycle management, quality assessment and preservation. Ratio of stable research objects vs. decayed ones over time.
Achievable	Availability of technological support for the management of research objects. This technology will be leveraged and customized to the case of research objects in ES.
Relevant	Providing the appropriate interfaces for managing, sharing and preserving experiments as research objects is crucial for the adoption of this concept by the community, and to foster the reuse of experimental results. Providing measures for assessing the quality of research objects will further support the publication of high-quality scientific results.
Timely	Technological support to be available at the end of the first year, ready for integration in the VRE. A second iteration will be released in M24.

The third SMART objective is to stimulate research objects sharing and reuse in Earth Science. In general, this deliverable has been designed around this premise. The semantic search and recommendation tools have as main objective to increase the research object findability so that they can be easily retrieved and reused by other researchers, supporting collaboration. In addition, quality has been identified as a factor that affects the potential reuse of research objects, and therefore this deliverable includes the design and implementation of checklists to assess the research object quality according to the requirements of the research communities participating in EVER-EST. Finally, the use of DOIs allow researchers to get credit when their work is reused and therefore be aware of their research impact. This supports collaboration by allowing the release of research objects as ways to convey research findings, even data, software, methods and intermediate results, to the corresponding scientific communities in addition to conventional publishing mechanisms like scientific papers.

SM_OB#3.3	**Stimulating sharing and reuse of research objects in Earth Science. Keeping track of the impact for prioritizing assignment of resources and observation time by data providers.**
Measured by	Number of research objects that have been found and reused, based on the recommendation functionalities provided by the system. Number of citations and acknowledgement of specific research objects as opposed to traditional publications in pdf format.
Achievable	Visual metaphors for the recommendation of research objects based on user preferences, profiles, and similarity that are available are adapted to meet the needs of the Earth Science community.
Relevant	Earth scientists will be enabled to find, reuse, and share relevant pieces of research investigations represented as research objects. This will increase the pace of scientific development in the area and minimize time and effort for subsequent work developed on top of existing one. The analysis of the impact generated will allow prioritizing observation time and resources for generation and maintenance of observation data.
Timely	Technological support to be available at the end of the first year, ready for integration in the VRE. A second iteration will be released in M24.

1.3 Overview of the document

Rather than listing the components, the document presents them embedded in sections that corresponds to the project objectives related to research objects. First section 2 "***Research Object Components for Earth Science***"

contextualizes the tools described in this deliverable in the VRE architecture. Next, section 3 "***Components for Search and Recommendation*** " describes: i) the current search system that has been adapted to the research object needs previously elicited from Earth Science disciplines through the EVER-EST VRCs, ii) a new version of the search system where semantic technology is used to process the research object content and produce a more intelligent search engine aware of such content and not only of explicitly defined metadata, and iii) the recommender system designed using the Collaboration Spheres metaphor so that researchers are enabled to carry out explorative search tasks. Section 4 "***Components for the Preservation and curation of Research Objects*** in Earth Science" describes the design and implementation of checklists from the requirements elicited from the research communities. Section 5 "***Components for Enabling Research Objects as Scholarly Communications***" presents the DOI system and its impact in the research object lifecycle and scholarly communications. Finally, section 6 "***Conclusions and Future Work***" present the next steps to integrate some of these tools in the VRE.

2 Research Object Components for Earth Science

This section presents the research object components in the context of the EVER-EST technical architecture generated in D5.1 “VRE Architecture and Interfaces Definition”. This architecture comprises different layers that reflect the shared purpose of the components contained in each layer. The EVER-EST architecture separates data access from their processing and presentation. There is a data layer, a service layer and a presentation layer. Figure 2 depicts a simplified view of the EVER-EST architecture where the focus is on the components that are related to the research objects described in this deliverable. In orange, the components covered in the current deliverable. Note that this figure only depicts two of the architecture layers: the presentation layer is the VRE portal, and the services layer represented by the RO services box.

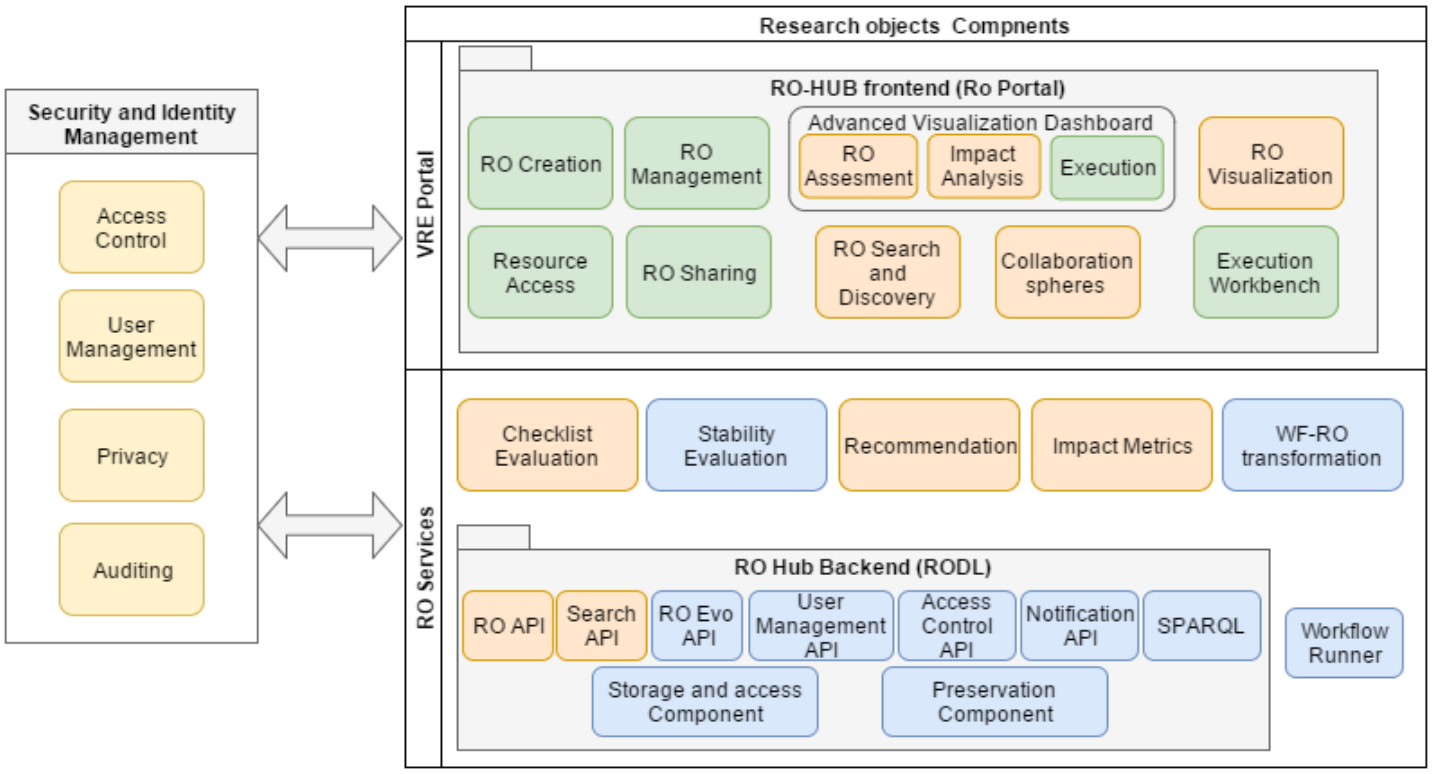

Figure 2 Research object components in EVER-EST general architecture.

The first tools described in this document are the semantic search engine and the recommender system. The search engine is built using the presentation layer component “RO Search and Discovery” - allowing users to pose queries, and visualize and browse the search result -, and the “Search API” in the service layer, indexing all research objects stored in ROHUB. The recommender system in turn uses the “Collaboration Spheres” component in the presentation layer to define the recommendation context, and displays the recommended items obtained through the “recommendation” component in the service layer.

The next tool to address are the checklists that have been designed following the VRC requirements regarding research object quality. The checklist functionality includes the “RO assessment” visual component that interacts with the “checklist evaluation” component in the service layer.

Finally, DOIs, the mechanism used to identify research objects in the scholarly publication context, affect the research object lifecycle, and hence the RO API that supports this lifecycle was modified. The visualization

components affected by the use of DOIs were the RO Visualization component and the research object "Impact Analysis".

2.1 Transversal aspects: authentication and privacy

Users have the possibility to set their research objects as private or public, and this privacy information must be taken into account by the retrieval and visualization tools, checklists management and DOI assignation processes. EVER-EST architecture, defined in deliverable D5.1, supports research object privacy through the authentication and privacy components that are part of the security and identity management module (see Figure 2).

The search and recommender services can be used by authenticated users or not. If the user is authenticated, then the search results will include user private information. In case of an unauthenticated user, the search and the recommendation results will automatically exclude all research objects that are marked as private.

On the other hand, users must be authenticated in the platform to assign checklists in order to evaluate the quality of their research objects. Nevertheless, the checklist evaluation results for public research objects are available to all users, and for private research objects, results are available to the owner.

Finally, users must be authenticated in the platform to assign DOIs to their public research objects.

3 Components for Search and Recommendation of Research Objects in Earth Science

Searching and recommending research objects are challenging tasks since research objects are complex objects that potentially can aggregate multimodal pieces of information such as documents and presentations, numerical datasets, workflows and pieces of programming code, images and videos. In fact, the research object model does not impose any restriction to the types of content that may be included in a research object beyond being a local file or any web resource that is identified through a uniform resource identifier. Given the multimodal nature of research object content and the distributed feature of some of the aggregated resources, it is necessary to define the scope of the Information Retrieval (IR) and recommendation components in terms of the type of information they leverage in their internal process. In EVER-EST the retrieval and discovery processes are based on the research object metadata and the text that can be extracted from the aggregated resources. If present, the research object metadata is the primary source of information regarding the content. However sometimes such metadata is not exhaustive, e.g. authors may not specify the research object description or provide a descriptive title, thus hampering retrieval and recommendation. In such cases, the retrieval and recommender system tap into the research object textual content that is extracted from plain text files, Ms Word and PDF documents, and PowerPoint presentations.

This section is structured as follows. First the search system description is presented in section 3.1. Including general concepts regarding the design and implementation of modern search engines (section 3.1.1), the metadata-based search system available in ROHUB (section 3.1.2), and the semantic search engine that leverages RO content (section 3.1.3). Next, the recommender system is described in section 3.2 where two different and complementary text processing techniques are being used: Explicit semantics (3.2.1.1) and word embeddings (3.2.1.2).

3.1 Research object search

Current keyword-based search functionality in ROHUB is based on a statistical processing of terms that appear in the research object metadata (see Figure 3). Nevertheless, this search engine has some drawbacks mainly due to the lack of semantics when processing natural language and the fact that users may generate incomplete metadata. Therefore, in EVER-EST a new semantic search engine is being developed to ameliorate the shortcomings of the current system [9]. The semantic search engine leverages both research object metadata and content as source of information and carries out an intelligent processing of texts so that it can understand the semantics conveyed in the research object content.

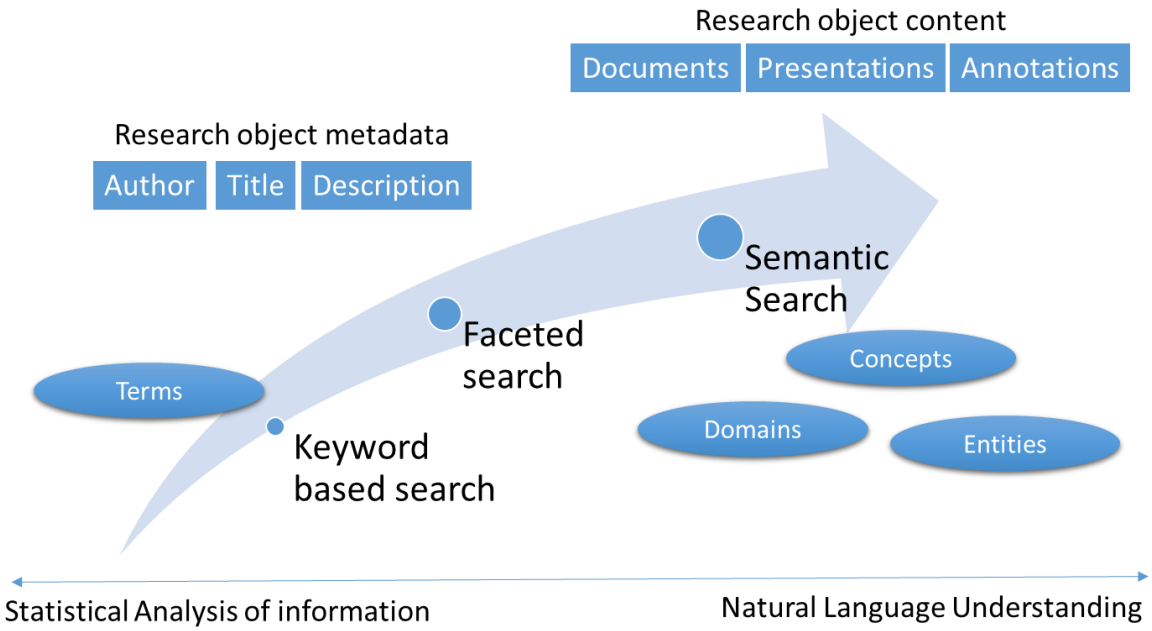

Figure 3 Search engine approaches for research object retrieval

The implementation roadmap foresees that both search engines will be integrated and a unique search point will be offered to the end user.

3.1.1 General concepts related to search engines

In a general context one of the most used models to deal with text in Information Retrieval (IR) processes [1],[2] is the **bag-of-words model.** In this model, documents are seen as sets of words where word order is not important, and therefore disregarded, and word importance in the document is defined with a numerical value. This simplified representation makes it feasible to process computationally large document collections. Theoretically, the bag of words comprises words, but in practice, it contains terms. The terms definition is wide and it includes words, key words, compound names, or n-grams. Similarly, a document definition is broad since it only requires being a text resource, hence documents could be sentences, paragraphs, code comments, keywords in queries, documents, or aggregation of documents.

In addition to the document representation model it is required a robust model to operate on text documents represented according to the bag-of-word model. The **vector space model** is an algebraic model where documents are represented as vectors in a multidimensional space, and therefore all vector operations are available to process text documents. The multidimensional space size corresponds to vocabulary size, i.e., the number of distinct words used in the document collection. Of special relevance for search and recommender systems is the cosine function that is used as similarity measure between documents. The main assumption is that the more similar the angle of the vectors, the more similar the corresponding documents. The cosine function tends to 1 if the angles of the vectors are similar and to 0 otherwise, and it can be used to generate rankings of similar documents. Precisely this is what is required in IR where given a query the system should return a ranked list of similar documents.

A document vector explicitly indicates the presence or the lack of a term in the collection vocabulary by assigning a numerical value or weight. This weight is also useful for indicating how relevant or important the term is for the document and the retrieval tasks. Term Frequency – Inverse Document frequency (**TF-IDF**) is a popular weighting scheme that aims at measuring how important a term is to a document in a collection. TF-IDF prioritises frequent

terms in the document that are not so frequent in the document collection. The result is that highly frequent terms such as preposition and articles are not considered important since they are also highly frequent in the other documents in the collection. This feature makes TF-IDF popular in IR since it helps to decrease the importance of stop words; i.e., words that do not help to discriminate between documents.

Finally, the last issue to consider when designing an IR system is the data structure used to implement a successful vector space model. For large document collections this data structure is a main design decision that affects the deployment of the system. In an information retrieval system, the main goal is fast query speed and therefore a good data structure should support fast queries. The straight implementation of a vector space model is a **m** x **n** matrix, where m is the number of documents and n is the vocabulary size. Searching for documents containing a term is very fast since the system only has to look up for the term column and retrieves the documents in the rows. However, for large document collections both the vocabulary size and the document collection are big numbers and therefore it is highly probable that the matrix does not fit in memory.

Other option is to use a forward index, i.e., an index that maps documents to the terms that appear in them. This data structure saves memory since the system only stores the terms that appear in the document leaving out the potentially large set of vocabulary terms not use in the document. However, forward index lookup speed is very low since the system has to traverse the entire index, document by document, to find which of them contain a term. The most suitable data structure for implementing the vector space model is a **reverse, sometimes called inverse, index** that maps terms to the documents that they appear in, along with other statistics useful for the selected weighting scheme. Querying a reverse index for documents containing a term is straightforward since terms are the index entries that points to the documents.

3.1.2 Faceted and keyword search from metadata

The research object model specifies the vocabulary and relations for capturing and describing research objects, their provenance and lifecycle. The model is formalised as a network of ontologies (see D4.1) including a core ontology, which provides the basic structure for the description of research objects, their aggregated resources and related annotations; plus extensions for describing evolution aspects and experiments involving scientific workflows. Moreover, the RO model implements a set of guidelines concerning the recommended properties and relations for the annotation of research objects and their aggregated resources through additional extensions and vocabularies (see D4.2).

More technically, a research object is specified in the RO model as an ORE aggregation described by a manifest resource, i.e., an RDF description of the content and the structure of the specific research object (see Figure 4). The research object aggregates a set of resources, which are referred to and linked to it by means of their URIs, and a set of annotations about a specific research object and the resources it aggregates. Annotation (expressed with the Web Annotation ontology) describe the link between a target resource (here aggregated in the research object), and a body resource, which is typically provided as a separate RDF graph (except from core annotations as creator and creation date that are in the manifest) in order to describe and relate each individual resource (see Figure 5). The annotation body comprises a description of the target resource in the form of a set of RDF metadata. Regardless of where the annotation is created (manifest or separate RDF graph), it translates into an RDF statement in the form (subject, predicate, object), where the subject is the target resource, the predicate is the annotation property, and the object is the value. These annotation properties are specified by RO core ontology extensions, mainly roterms and roes. These two ontology include both definition of terms, and reference to terms from well-known vocabularies, which are useful for the annotation of resources in any scientific domain (roterms) like title, description, creation date and creator, and more specifically in the earth-science domain (roes) like geospatial and time metadata, or detailed intellectual property rights metadata.

```
<> a ro:ResearchObject ;
    ore:aggregates <helloworld.t2flow> ;
    ore:aggregates <artifact/hello> ;
    ore:aggregates <ann1> ;
    ore:isDescribedBy  <.ro/manifest.rdf>
    dct:created "2011-12-02T15:01:10Z"^^xsd:dateTime ;
    dct:creator [ a foaf:Person; foaf:name "John Doe" ] .

<ann1> a ao:Annotation ;
       a ro:AggregatedAnnotation ;
       a ore:AggregatedResource ;
       ao:body <123.rdf> ;
       dct:created "2011-12-12T15:01:10Z"^^xsd:dateTime ;
       dct:creator [ a foaf:Person; foaf:name "John Doe" ] ;
       ro:annotatesAggregatedResource  <> .
```

Figure 4. RO manifest example

```
<ro> dct:title "RO Title"^^xsd:string .
```

Figure 5. RDF annotation body example

The way the RO model enables to associate annotations to research objects, along with the annotations properties specified by the RO model extensions and vocabularies, provide the basis for the implementation of a faceted-based search interface in ROHUB. Faceted search (aka. faceted navigation or faceted browsing) is a technique for accessing information organized according to a faceted classification system allowing users to explore a collection of information by applying multiple filters[1]. The faceted classification system classifies each object along multiple explicit dimensions, called facets, which correspond to objects properties. In the case of ROHUB, these facets correspond to the annotation properties (specified by the RO model extensions) associated to a research object.

Internally, these RO annotation properties are indexed in a Solr server, which is then used as the information source for the facets filters displayed in the interface. Additionally, Solr indexes two calculated properties, the number of resources aggregated by the research object and the number annotations associated to the research object. These two properties are derived from the core RO description. The interface shows a selected subset of these properties for the users to navigate and search research objects. This subset represents those properties considered as the most relevant for the users.

As part of EVER-EST project, a completely new ROHUB portal is being developed following a modular approach in order to facilitate the integration of components in other portals, to improve the usability and performance, and to update the technologies used in the implementation. As depicted in Figure 6, the faceted filters list will include more facets in order to facilitate the navigation and search based on multiple criteria. Note, however, that in order to make this interface useful, it will require that research objects are appropriately annotated with all of these properties. For the upcoming release of ROHUB existing research objects will be curated and users encouraged to provide quality metadata in order to make their research objects publicly available. More details about the new version of the ROHUB portal are presented in D5.4

Another possibility for searching research objects is by searching keywords in its metadata information. This simple mechanism allows finding quickly research objects having some specific keywords in any of their annotation properties (e.g., title, description, creator, etc.).

[1] https://en.wikipedia.org/wiki/Faceted_search

In ROHUB this component is implemented as a search text box, where the users enter the keywords they would like to search in the annotation properties (Figure 7). Internally, this component also uses Solr to find matches of the words entered in the search text box inside the values of the research object annotations. The search engine in the new ROHUB portal implements an optional field to select the research area related to the research object. The results are displayed in the faceted search results list, and the faceted filters list is updated accordingly.

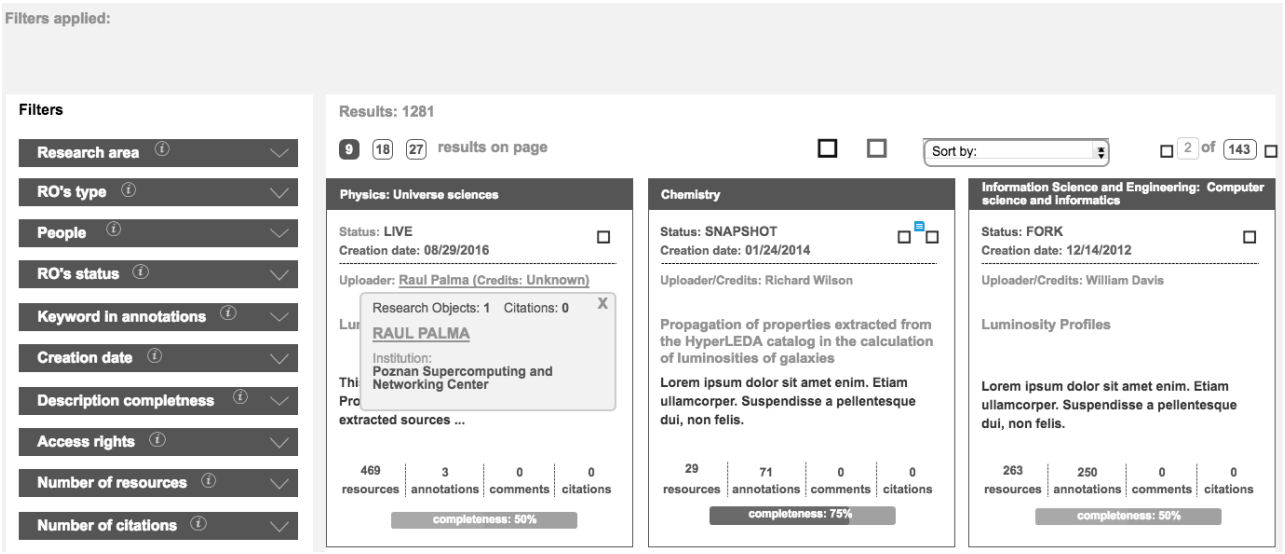

Figure 6. ROHUB faceted search-based interface (v2) – tiles view

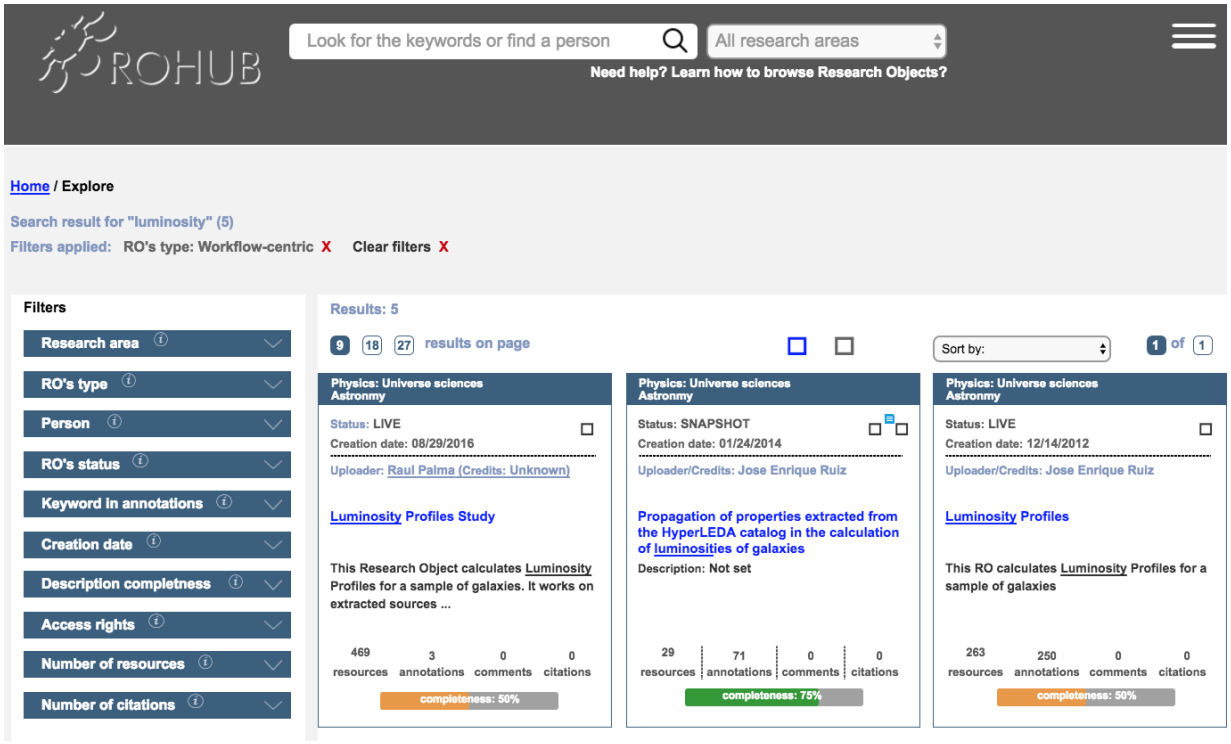

Figure 7. Keyword-based search in ROHUB (v2)

3.1.3 Semantic search from content

Despite being based on standard search technology the keyword and faceted search previously described have two main limitations due to potential absence of metadata that search engine indexes, and the inherent weakness of the bag-of-words models. On the one hand, current search functionality taps into ROHUB metadata about the research object. However, part of these metadata is user-generated and might be incomplete or not present all. In these cases, the search engine would miss the metadata it needs and therefore it would be unable to provide access to these research objects.

On the other hand, bag-of word models fail to capture semantic relations between words. For instance, it is not possible to relate two documents if they do not share words, even if some of these words are **synonyms**. The lack of synonyms in an IR system can result in a **low recall of the search results** since documents containing the synonyms and not the query terms are not included in the search results. Furthermore, bag-of-word models treat words as simple character strings disregarding the word meaning. This lack of word semantics **decreases search engines precision** when they are queried with **ambiguous words** since systems are unable to discriminate documents where the word has being used in the user intended meaning.

A semantic-aware search engine that exploits research object content has been developed to avoid the aforementioned limitations. This search engine indexes metadata and the text extracted from the research object aggregated resources. In addition, in contrast to bag-of-word models that treat words as character strings the search engine works with concepts that group words according to their meaning, and these concepts are related between them according to semantic and linguistic relations, as defined in an existing knowledge graph. Finally, in addition to the user-generated metadata the semantic processing enriches the description with metadata about the content that is generated automatically. This content-based metadata includes domain, named entities mentioned in the text, and main concepts, among others. Cogito, Expert System's semantic intelligence platform (http://www.expertsystem.com/cogito) is used to analyse research object content and obtain the associated semantic annotations.

3.1.3.1 Deep semantic analysis using Cogito

The core component of Cogito is the **sensigrafo, a semantic network where knowledge is represented as a graph of meaning of words, i.e., concepts, and relationships between concepts**. Inside Cogito, lemmas are not organized in alphabetical order, such as in a dictionary, but in **syncons** or groups of synonyms that represent the same meaning or concept that the lemmas express. Each syncon is a node in the semantic network that is linked to other nodes through semantic and linguistic relationships in a hierarchical structure. In this way, each node, in addition to its meaning and attributes, is enriched by the characteristics and meaning of the nodes that are above (supernomen, in Cogito terminology) or below it (subnomen). Each of these links identifies a kind of relationship that links the concepts in a language, organizing the concepts in the semantic network. For example, concepts are organized starting from the less specific to the more specific (*vehicle/car/SUV*) or as potential subjects or objects of a verb, etc.

Syncons can contain lemmas which are words and collocations. The main elements of each syncon are: grammar type, semantic link, the definition/meaning, the domain, and the frequency. In the semantic network, the real meaning of each syncon is a combination of its elements (**synonyms**) and the relationship between the syncons. Cogito's semantic network includes different kinds of links between the syncons, which results in a greater ability to represent the understanding of a language. For a superior understanding of text content, Cogito uses a **morphological analysis** to process keywords, a **grammatical analysis** that understands the base lemma, and a **logical analysis** that identifies the parts of speech in a sentence (subject, verb, object, preposition, etc.). In addition, it performs a **semantic analysis** that disambiguates the meaning of the words by reusing the information of the previous analysis.

Cogito is composed of several integrated elements that are used to disambiguate texts and process natural language, which is essential for the automatic comprehension of a text. **Cogito parser** identifies the single elements that constitute a text, assigning them their logical and grammatical value. Consider the examples: *(a) there are 40 rows in the table. (b) She rows five times a week.* While traditional systems would treat the use of *rows* equally, Cogito understands their different grammatical role in each sentence, and therefore the different meaning of each. Recognizing a word independently of its written form is important. Cogito distinguishes gender—masculine/feminine—and number—singular/plural—in words and correctly associates all forms of verbs to their common meanings instead of simply identifying them as different words, as other systems do.

Cogito disambiguator analyses single sentences or entire documents to distinguish the precise meaning for each word from all the various meanings associated with a term, eliminating any possible ambiguity. This way, Cogito can understand the meaning of words in a specific context, mirroring the way humans process information. During document analysis, Cogito uses a retention technique to determine the semantic context of a text that is used in the disambiguation and extract meaning with the best possible approximation

The outcome of Cogito's semantic processing is a cognitive and conceptual map—essentially a structured representation of previously unstructured text:

- Each concept expressed in the text is uniquely identified regardless of which words are used to represent it in the text analysed.
- Each agent is associated with the action carried out.
- Each object is connected to the related action.

In this representation of content, a document's main topic, as well as other topics, dates, numbers and other meaningful information, are identified and stored. The map provides a document with structure, supporting formal text processing tasks such as indexing, classification, summarization and translation.

3.1.3.2 Semantic search

This section describes the design and implementation of the semantic search engine. Figure 8 depicts the main stages and processes of the semantic search and shows how research object data and metadata are processed semantically to produce the search index that is used to answer user queries. In addition, the figure depicts the technologies used to support the different processes stages.

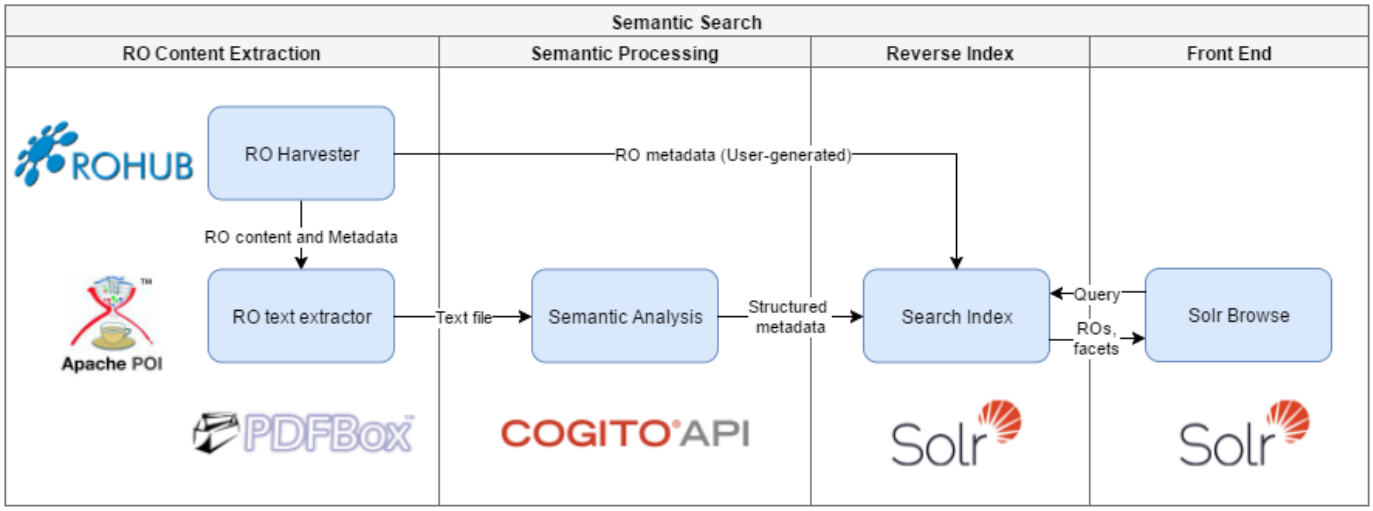

Figure 8 Semantic search components

Research object content extraction

In this stage the goal is to collect metadata and content to produce a single text file containing the textual content. The first process is the **RO Harvester** that gets each research object zip files from ROHUB, plus the metadata that is not present in the manifest (title, creation date, author, status, and description). ROHUB RESTful API is used to retrieve the following information:

- rodl/ROs/ : List of stored research objects
- rodl/zippedROs/ : Zipped research object file
- portal/sparql : research object metadata

Next, the **RO text extractor** analyses the manifest and identifies, according to the research object vocabulary, all resources that are documents, bibliographic resources, hypothesis, research questions, conclusions, and papers. From this resource set, and with support of specialized software libraries, text is extracted from plain text files, PDF, Word documents and PowerPoint presentation. Apache PDFBox (https://pdfbox.apache.org/) is used to extract text from PDF documents, and Apache POI (https://poi.apache.org/) is used to extract text from Word documents and PowerPoint presentations. All the text collected from the aggregated resources plus the title and description are merged in a single text file.

Semantic processing

Cogito API is used to semantically annotate the text that aggregates all the textual content in a research object. Cogito applies the morphological, grammatical, logical and semantic analysis and returns an xml file with the main lemmas, main syncons, main word groups, main domains, and the named entities, including people, places and organizations that are identified in the text.

In the following an example is presented where an existing research object is processed with Cogito to show how raw text is transformed into structured semantic information. In this example, a research object entitled SeaMonitoring01-snapshot-1 is used (see Figure 9). Note that this research object lacks of title and description and therefore retrieving it with the regular search is almost impossible unless one knows its URI.

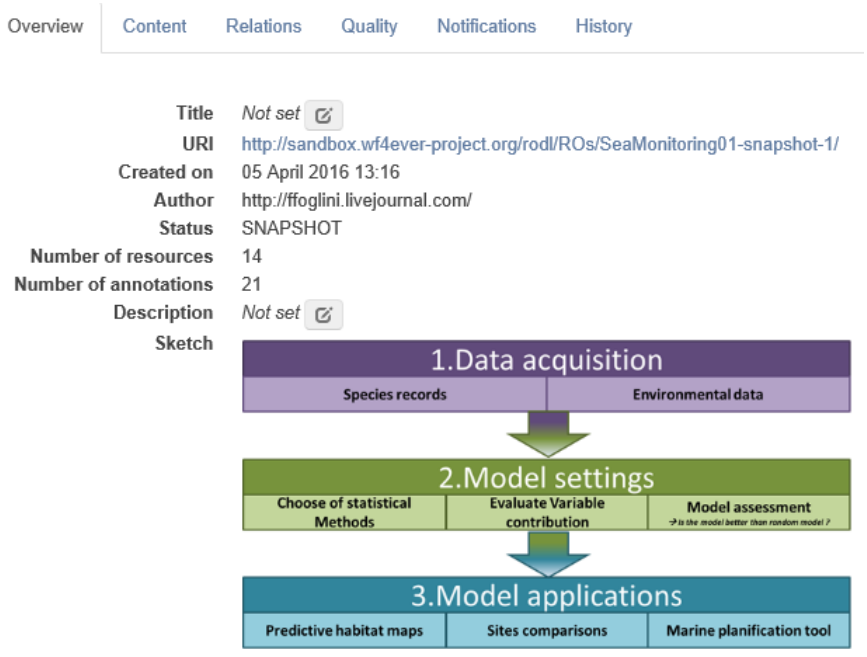

Figure 9 Sea Monitoring research object processed with Cogito

The research object content was processed through the RO Harvester and RO text extractor procedures, and the text content fed into Cogito. Cogito yields the following metadata summarizing the RO textual content. First named entities are listed: people, organizations and places. For people, see Code Snippet 1, Cogito identifies the name, gender, aliases and position (track element) in the text where the same person was mentioned. Cogito also indicates if the person plays a role in another entity. The example shows that this person is an employee of the "Ifremer Mediterranean Center".

```
<ENTITIES TYPE="PEOPLE">
  <ENTITY NAME="Annaelle Bargain">
    <PROPS>
      <PROP NAME="ALIAS" VALUE="Bargain"/>
      <PROP NAME="SEX" VALUE="F"/>
      <PROP NAME="SURNAME" VALUE="Annaelle Bargain"/>
    </PROPS>
    <RELS>
      <REL NAME="humanrole" SYNCON="44906" VALUE="employee">
        <REL ESYNCON="17338" NAME="humanrole_orgspec" VALUE="Ifremer Mediterranean Center"/>
      </REL>
    </RELS>
    <TRACKS>
      <TRACK BEGIN="29" END="44"/>
      <TRACK BEGIN="1011" END="1026"/>
    </TRACKS>
  </ENTITY>
```

Code Snippet 1: People recognised in the text by Cogito.

Organizations mentioned in the text are listed as illustrated in Code Snippet 2. Similarly to the people case, Cogito recognizes the organization name, the aliases used in the text, and the positions where it was mentioned.

```
<ENTITIES TYPE="ORGANIZATIONS">
  <ENTITY NAME="Arctic Intermediate Water">
    <PROPS>
      <PROP NAME="ALIAS" VALUE="AIW"/>
    </PROPS>
    <TRACKS>
      <TRACK BEGIN="681" END="708"/>
      <TRACK BEGIN="739" END="745"/>
  </ENTITY>
```

Code Snippet 2 Organizations recognised in the text by Cogito

The last entity type is location (see Code Snippet 3). Cogito recognizes the entity names, the aliases, the syncon that represents the location in the semantic network, and the hierarchy in the geographical taxonomy where the location is classified.

```
<ENTITIES TYPE="PLACES">
  <ENTITY NAME="Shetlands">
    <PROPS>
      <PROP NAME="ALIAS" VALUE="Shetland"/>
      <PROP NAME="ALIAS" VALUE="Shetland Islands"/>
      <PROP NAME="SYNCON" VALUE="12625640"/>
      <PROP NAME="GEOREF" VALUE="Scotland/United Kingdom/Europe"/>
    </PROPS>
    <TRACKS>
      <TRACK BEGIN="15170" END="15185"/>
      <TRACK BEGIN="15262" END="15269"/>
    </TRACKS>
  </ENTITY>
```

Code Snippet 3 Location recognised in the text by Cogito

In addition to entities Cogito summarizes the main lemmas, word groups, concepts or syncons, and domains used in the document. Code Snippet 4 shows the metadata identified in the Sea Monitoring research object. Syncons 41320 and 414815 refers to *seabed* and *mound* concepts.

```
<RELEVANTS TYPE="MAINLEMMAS">
  <RELEVANT NAME="seabed" SCORE="2.70"/>
  <RELEVANT NAME="mound" SCORE="2.10"/>
</RELEVANTS>
<RELEVANTS TYPE="MAINGROUPS">
  <RELEVANT NAME="seabed mound" SCORE="3.20"/>
  <RELEVANT NAME="mound structures" SCORE="1.40"/>
  <RELEVANT NAME="seafloor characteristic" SCORE="1.40"/>
</RELEVANTS>
<RELEVANTS TYPE="MAINSYNCONS">
  <RELEVANT NAME="41320" SCORE="13.30"/>
  <RELEVANT NAME="41485" SCORE="11.20"/>
</RELEVANTS>
<DOMAINS TYPE="RELEVANTS">
  <DOMAIN NAME="hydrography" RANKING="1" REFS="138" RELIABILITY="13.00" SCORE="6.40"/>
  <DOMAIN NAME="geology" RANKING="2" REFS="65" RELIABILITY="6.00" SCORE="3.40"/>
</DOMAINS>
```

Code Snippet 4 Main lemmas, word groups, syncons and domains recognised in the text by Cogito

Note that these annotations provide a concise, structured representation of the research object content and therefore constitute a valuable source of information that the search engine can leverage to offer an overall better search experience.

Search index

At this point the collected information comprises the user-generated metadata that authors define when loading research objects to ROHUB, and the metadata automatically generated by cogito from the research object content. All this metadata must be turned into documents and fields that are the entities that search indexes understand.

Each research object is considered as a document and the metadata as fields. The document unique key corresponds to the research object URI. Table 1 shows the document fields and the corresponding element in Cogito's output. All the fields are indexed, stored, and multivalued. Indexed means that they are used in queries to retrieve matching documents, stored means that the field is retrievable in a query result, and multivalued means that a document can contain many values for this field. A Content field of type text is created to aggregate all the other fields including the document id. The content field is used as primary search source when the user does not specify the field name that he wants to query. All fields that are being considered for faceting are of type String. The Concept IDs is an integer field, and the words and compound terms that are not included in the facets are text

types. The different between String and Text fields is that the former are not tokenised nor analysed while the latter goes through these stages.

Table 1. Mapping between cogito elements and index fields

Field Name	Cogito Synthesis element	Field type	Facet
PEOPLE	PEOPLE	String	Yes
ORGANIZATIONS	ORGANIZATIONS	String	Yes
PLACES	PLACES	String	Yes
WORDS	MAINLEMMAS	Text	No
COMPOUND_TERMS	MAINGROUPS	Text	No
CONCEPT_IDS	MAINSYNCONS	Integer	No
CONCEPTS	-	String	Yes
DOMAINS	DOMAINS	String	Yes
CONTENT	All	Text	No

All the text fields are analysed at indexing time with the standard analyser that roughly splits the text using spaces and punctuation marks. The standard analyser recognizes email addresses and internet host names. No filters are applied once this task is complete. At querying time all the text fields are analysed with a custom analyser that uses Cogito to process the user query and obtain the tokens from Cogito output. The index is implemented using Apache Solr (http://lucene.apache.org/solr) .

Querying the Index with research object semantic information

An implementation of the semantic search engine for research objects is deployed and available for general use at http://everest.expertsystemlab.com/browse. This web application includes different features that enhance the search process and experience including facets, research object content summary, autocomplete, and *more like this* to find similar items. Note that although these features are regularly part of search engines, in this case all of them are enriched with semantic information extracted from the research objects content. Figure 10 shows a screenshot of the search engine result page.

The semantic search engine has a search box where users type the terms that make up the query. In the search box, users can constraint their query to one of the semantic fields of information that were extracted with Cogito, i.e., Domain, Concepts, Expressions, Places, Organizations and People. In Figure 10 the user states that Geology is the domain of interests**. Note that the word *Geology* might not be mentioned at all in the retrieved research objects. However, Cogito semantic processing makes it possible to identify the domain of work in these research objects by means of the context described in the related text**. In addition, this search box has an autocomplete functionality (see Figure 11) that suggests terms related to the lists of concepts, domains, and expressions that appear in all the research objects indexed by the search engine. When a user selects one of the autocomplete suggestions, the search results are more precise since the content of the retrieved research objects is about the suggested term. Once again retrieved research object may not contain the suggested term explicitly since domains and concepts are broader entities that are described by many different terms.

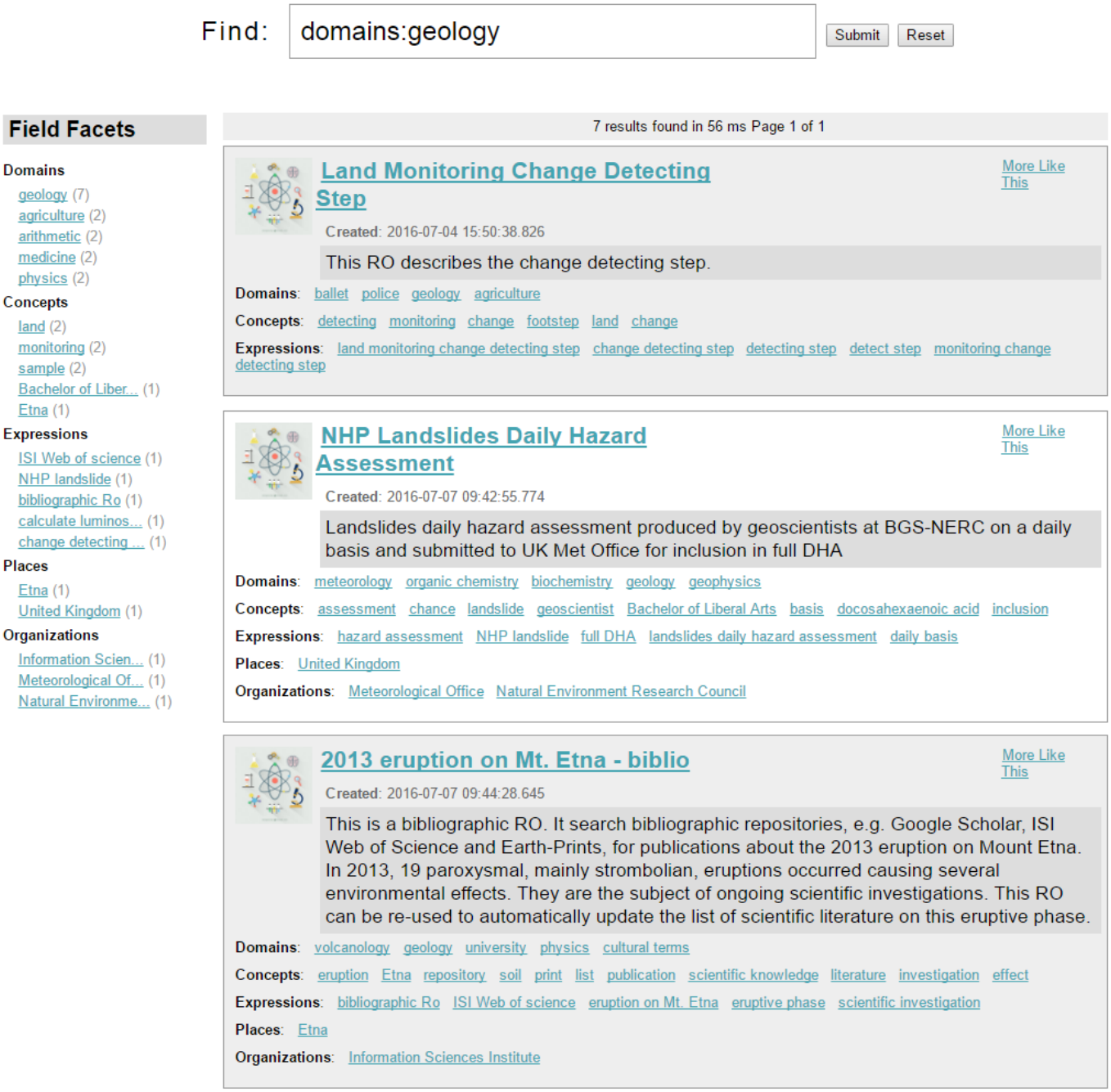

Figure 10. Overview of the semantic search engine

In the result page, research objects are summarized and classified into facets using the semantic information elicited from their content. Each research object summary contains the research object explicit metadata such as thumbnail sketch, title and description, and the semantic metadata including domains, concepts and common expressions (compound words) identified by Cogito. In addition, both facets and semantic entities in summaries are clickable elements in the user interface that trigger new searches. Facets keep the search context while the links in the summary are new searches that discard the current search context.

Figure 11. Search box autocomplete list

A *More Like This* link is included along each search result to retrieve similar research objects based on the domains, concepts and expressions used in the retrieved research object. This a very useful feature since with only one click the user is posing an advance search that otherwise should be specified in the search box using logical operators over the different semantic fields included in the search index. Figure 12 shows the result of following a *More Like This* link in the research object entitled "Land Monitoring Calibrating Step". In the resulting page, under the section *Similar items* appear the three most similar research objects: "Land Monitoring Coregistering Step", "Land Monitoring Terrain correcting step", and "Land Monitoring subsetting component".

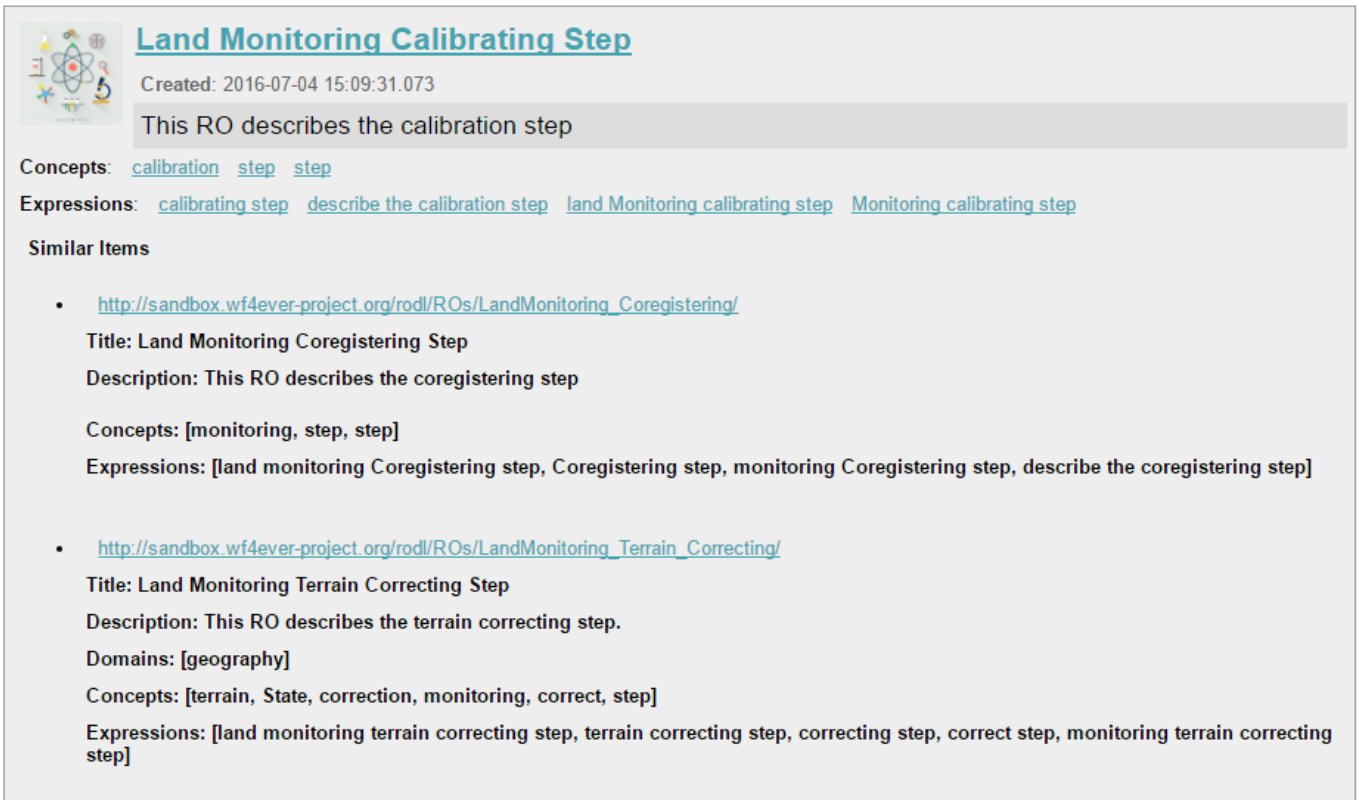

Figure 12. Similar research objects found by the *More Like This* option

3.2 Recommendation

The goal of the recommender system is to suggest research objects of interest to researchers exploiting the social dimension inferred from their collaborations and the domains of interest for their research. Thus, the recommender system follows a content-based approach enhanced with a social dimension. The social dimension refers to the social network that emerges from research collaborations such as co-authoring of scientific papers and in this specific case of research objects. In EVER-EST the social dimension has been enhanced with the set of prominent people mentioned in the content.

Similarly to the semantic search engine, the recommender system processes content using semantics technologies that allow a better understanding of the main concepts found in research objects. Nevertheless, the recommender system goes a step ahead to overcome some limitations of semantic processing using a semantic network that are related to dynamic evolution of vocabularies in research domains that might not be covered in the semantic network. The recommender system uses, as a complement to the semantic network approach, word embeddings to represent words. Word embeddings are semantic representations of words, which are learned from the set of documents relevant in a domain, and therefore the vocabulary used in the document collection is completely covered in this approach.

The recommender approach tackles two of the main issues of content-based recommender systems: accuracy and diversity. Accuracy is a key factor when evaluating the performance of recommender systems since it guarantees that recommendations are relevant, while diversity of recommendations across topics or domain is perceive as of great value for users [3]. The semantic analysis of the research object content produces more accurate recommendations since the meaning of words are disambiguated and therefore the recommender system suggests research objects that are more related to the user interests, avoiding research objects containing unrelated word meanings. In addition, the improved social dimension helps to increase the diversity of recommendations by stretching the limits of the co-authors social network border to other authors mentioned in their research.

3.2.1 Social semantic recommender service

In general, a content-based filtering approach uses a list of features of an item to recommend other items with similar characteristics. In EVER-EST recommender system the users' research objects are used as reference objects with which the other research objects are compared based on common features that represents ROs content.

The first source of features regarding the research object content is the user-generated metadata gathered in ROHUB. The title and description can give clues about the content if present. However, these metadata are pieces of text and consequently the system faces the inherent challenges of comparing two pieces of text to find similar ROs we. The alternative to the user-generated metadata is to analyse the content to obtain the set of features. Nevertheless, as mentioned in section 3.1, analysing research object content is not a trivial problem due to the multimodal nature of the aggregated information.

In summary, the recommender system faces the fact that research object content is mainly described by text pieces contained in the research object metadata and in the aggregated resources. As discussed previously the most frequent models to represent text as structured data, bag-of-words, fails to capture the semantics of the information written in natural languages, and this lack of semantics affects negatively the performance of the systems that use these models. Broadly the approaches used to identify word semantics can be classified in two types: i) Explicit semantic approaches where words are associated with concepts in a knowledge base, ii) Word embedding approaches where words are mapped to a multidimensional space in the hope that words with similar meanings are close to each other in the projected space [4].

The EVER-EST recommender system plans to use both, explicit semantics and word embedding approaches so that a more robust solution can be delivered. The reason for this hybrid approach to bring semantics to the recommender system is based on the coverage limitations of the knowledge bases used in the explicit semantic

approaches. These knowledge bases can take the form of thesaurus or more advanced semantic networks like Sensigrafo, and contain precise information about word meanings, usually gathered by a team of linguists, domain experts and knowledge engineers or ontology practitioners. The trade-off for the high quality of the represented knowledge is a low coverage of the knowledge about emerging terminology in very dynamic domains or very specific vocabularies used in particular disciplines. In the EVER-EST case the vocabulary belongs to Earth Science domains and it is very likely that some of the scientific concepts and their relationships are not included in general-purpose knowledge bases. Therefore, to cope with the low coverage of crafted knowledge bases the alternative is to use word embeddings as complementary approach since these representations are dynamically learned from a document corpus. Word embeddings can be updated in a regular basis thus incorporating new and emerging words and concepts.

Apart from semantically representing research object content and obtain more representative features, the recommender system requires a model to filter similar research objects so that they can be presented as recommendations. The recommender system uses the vector space model where documents, i.e., research object content, are represented as feature vectors in a multidimensional space, and the cosine function can be used to calculate the similarity between two documents, as explained in section 3.1.1. The rest of this section describes the implementation of the approaches used to process semantically the research object content. Note that in the current version of the recommender system the explicit semantics approach is implemented. The plan foresees the implementation of a word embedding approach for the next version of the system that will be described in deliverable D4.4.

3.2.1.1 Explicit semantics

The explicit semantics category refers to Natural Language Programming (NLP) tools that map words to concepts in a knowledge base according to the context where the word was used. This mapping between words and concepts is the result of a deep analysis of text, including lexical, syntactic, and semantic analysis, that gradually add knowledge about words yielding as a result the word meaning. Cogito, Expert System's semantic software, is used to carry out the semantic processing of text. For a description of Cogito's capabilities see section 3.1.3.1.

From the text, Cogito produces structured data about named entities, including people, organizations, and places. In addition, it synthetises the content of the document by identifying main domains, main concepts, main lemmas, and main groups of words used in the text. This set of structured data regarding the research object content is used as the set of features over which the system can compare research objects and filter them according to their similarity. Similar research objects are then recommended to the author. Note that the research object metadata include the authors and therefore the recommendation can include not only similar research objects but authors working in similar domains.

3.2.1.2 Word embeddings

The next version of the recommender system, to be included in D4.4, will incorporate a distributed word representation for text processing where words or documents are mapped to dense vectors of real numbers. Distributed word representations are also known as word embeddings since an entire vocabulary is embedded into a relatively low-dimensional linear space [6]. The goal of the word embeddings is to capture attributional similarities between words. That is words that appear in similar contexts will be close to each other in the projected multidimensional space. Word embeddings group words that share semantics properties and these groups have been used successfully in NLP tasks. Recently neural networks have been used to train the embedding vectors over large text collections. In [5], Mikolov et. al. showed that embeddings trained with a recursive neural network (RNN) encode, besides attributional similarities between words, syntactic similarities between pairs of words.

Word embeddings have been used to establish semantic similarities of words. For instance, applying the cosine function as indicator of similarity the most similar words to Sweden are Norway, Denmark, Finland, and Germany, which are Scandinavia nations and north European countries. In addition, semantic analogies such as Brother is to Man as Sister to Woman can be generated with algebraic operations over vector representations. For instance, vectors operation results in Country and Capital semantic relations such as Germany is to Berlin as France is to X, where X can be computed by the operating on the vectors Berlin-Germany+France yielding a similar vector to that of Paris. In addition, these models are able to identify syntactic analogies such as quick-quickly and slow-slowly or big-biggest and small-smallest.

Word2Vec

To enhance the recommender system a specific implementation of word embeddings called word2vec [5] will be integrated. Word2Vec is a popular implementation that proposes two architectures namely continuous bag-of-words (CBOW) and skip-gram, for computing vector representations of words. In these architectures, the neural network language model is trained in two steps. First continuous word vectors are learned using a simple model and then the n-gram Feedforward Neural Net Language Model (NNLM) is trained on top of these distributed representations of words. All models are trained using stochastic gradient descent and backpropagation

Continuous Bag-of-Words Model: This architecture predicts the current word based on the context defined by the previous and subsequent words (see Figure 13). In this model at the input layer, N previous words and N subsequent words are encoded using 1-of-V coding, being V the vocabulary size. This input layer is then projected to a shared projection layer for all words so that all words get projected in to the same position. It is called bag-of-words since the word order does not influence the projection. In their paper, authors train the model by building a log-linear classifier with four subsequent and four previous words at the input, and the training criteria is to correctly classify the current word.

Continuous Skip-gram Model: This architecture tries to maximize the classification of a word based on another word in the same sentence (see Figure 13). The idea is to use the current word as an input to a log-linear classifier with a continuous projection layer and predict words within a certain range before and after the current word.

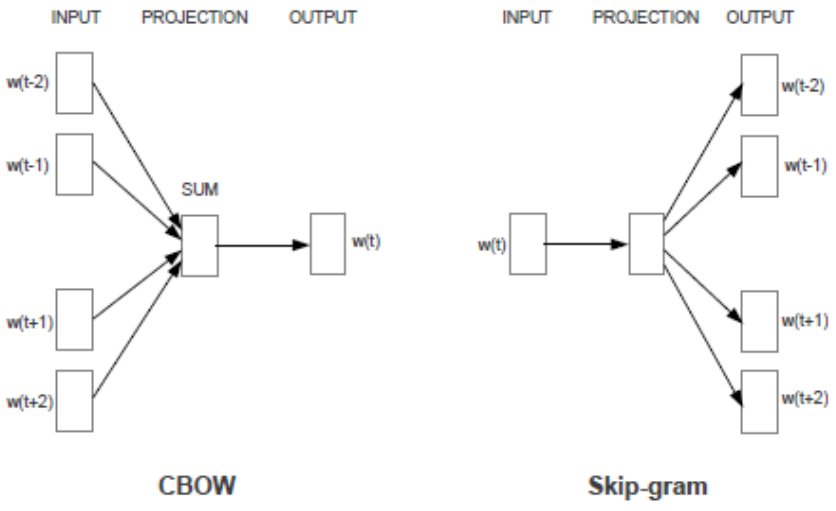

Figure 13 Continuous bag-of-words and skip-gram architectures

[Mikolov et al., 2013].

In the experiments authors reported that the CBOW works better modelling syntactic relations and the Skip-gram works slightly worst that the CBOW in the syntactic side and much better modelling semantic relations. Overall skip-gram model performed better than CBOW when averaging the number of correctly identified syntactic and semantic relations.

3.2.2 Collaboration spheres

Collaboration Spheres (CS) is the visual metaphor adopted by the user interface of the recommender system. Collaboration Spheres facilitate to explore, share and reuse research objects and user expertise based on the exploitation of semantic descriptions, relations and similarities between research objects and users. This metaphor has been used successfully as recommendation interface for research objects in a real-life scenario with myExperiment[2] data where users reported that they perceived that collaboration and reuse of scientific results was increased while the overhead of identifying and retrieving relevant scientific knowledge for a particular purpose was reduced. It has also been applied to enable serendipitous search and recommendation of possible reviewers for the submissions received by the journals edited by the American Psychological Association[3], one of the largest publishers in Psychology worldwide.

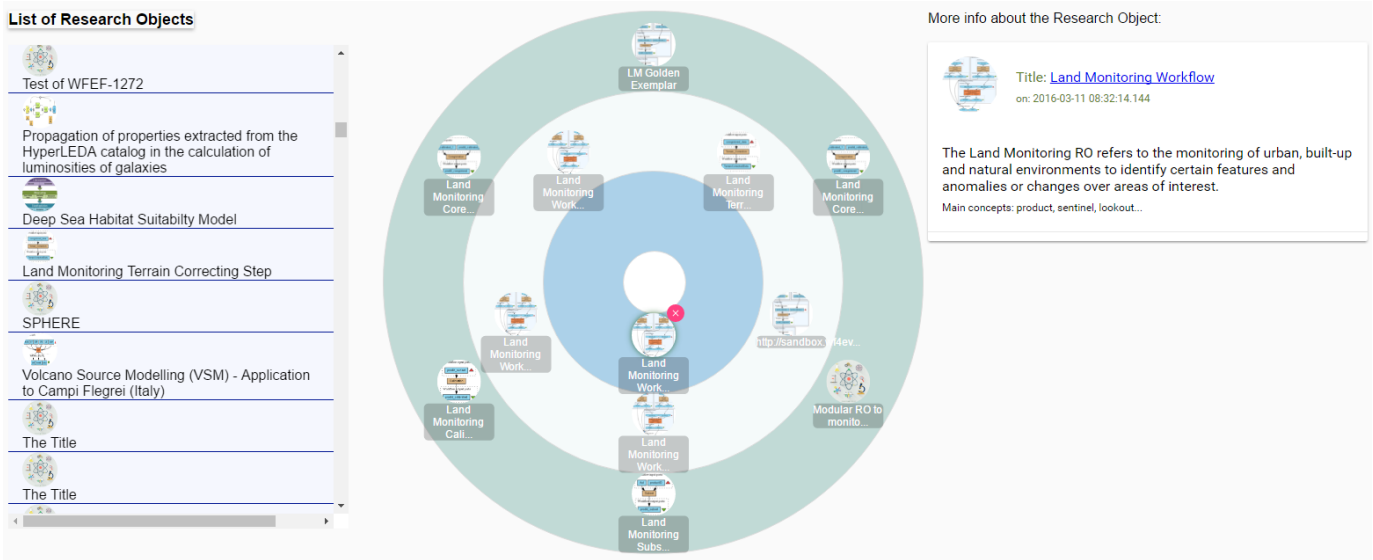

Figure 14. Collaboration Spheres interface

Figure 14 shows a screenshot of the Collaboration Spheres web application. In this first version, available at http://everest.expertsystemlab.com/spheres, the user interface presents a list of research objects, the Collaboration Spheres, and a summary card for research objects. The right panel, where the list of research objects is displayed, allows research objects to be dragged to the Collaboration Spheres so that they can be used as context to drive the recommender process. In this panel, research objects are clickable links that launch a summary card containing relevant research object information such as title, creation date, description and main concepts.

The spheres panel is used mainly to define the recommender context and to display the recommended items. It comprises a set of four co-centric spheres (see Figure 15) where the two smallest spheres are used to define the

[2] http://www.myexperiment.org/

[3] http://www.apa.org

recommendation context and the two outer spheres are placeholders for the recommendation results. Results in the third sphere are more relevant, and results in the fourth sphere are less relevant.

The recommender system is user-centric and hence the smallest sphere represents the user that is using it. The user can drag up to three research objects from the list into the second sphere to define the context of the recommendation. From each of these research objects the system extracts the main domains and concepts, which are used to perform the content-based recommendation.

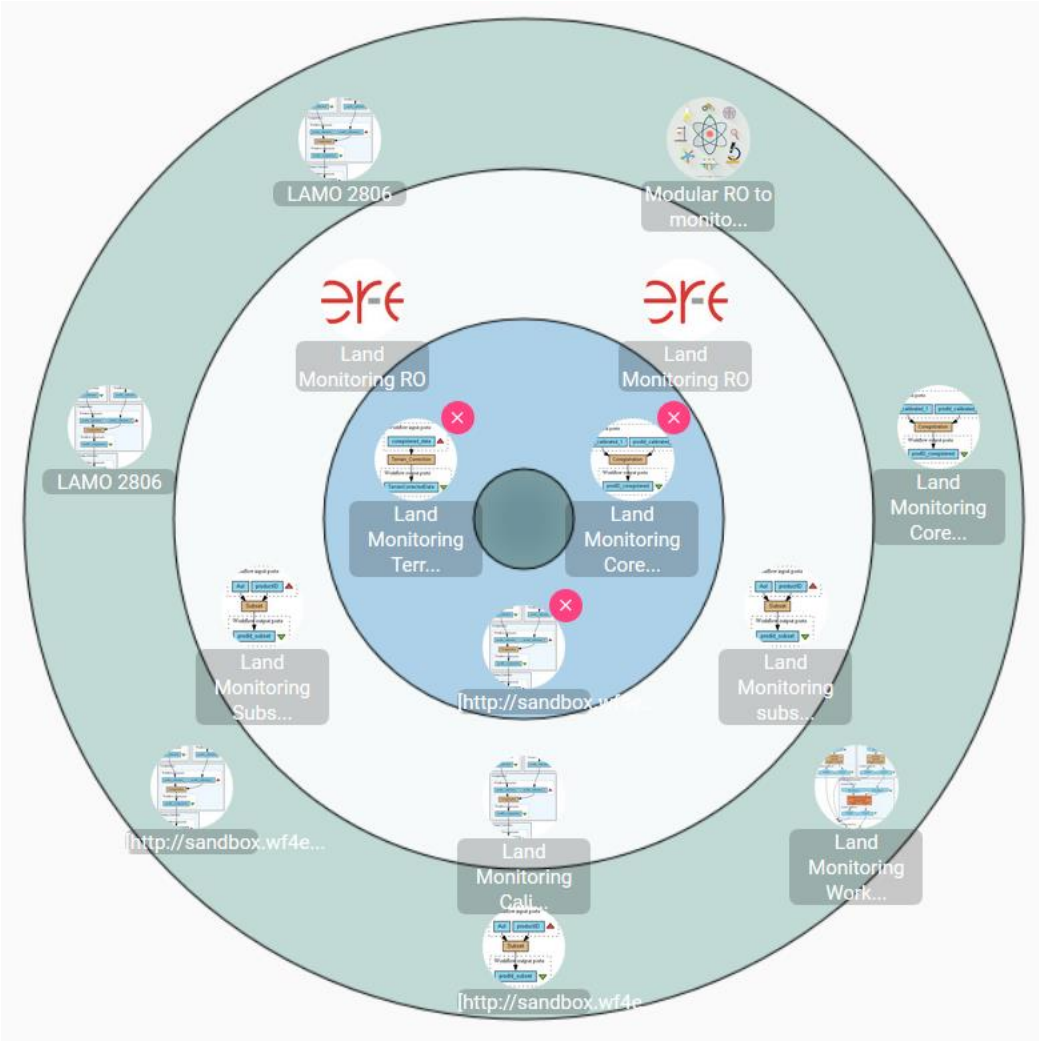

Figure 15. Detailed view of the spheres component

This release of the recommender system focuses on recommendation around research objects. Next release, to be delivered in D4.4, will also address user-centric recommendation and study the advantages of using it as a complement to research object-centric recommendation.

4 Components for the Preservation and curation of Research Objects in Earth Science

Ensuring long term preservation and curation of research data, processes, and methods is one of the main goals of the research object model. D4.1 introduced the metrics of completeness and stability as quantitative measures related to research object integrity and authenticity, two desirable features when it comes to research object preservation. In particular, completeness measures how complete is the research object information regarding the minimum information it is expected to contain, while stability measures the degree to which a research object remains functionally unchanged with respect to its specification and properties in the presence of changes on its resources. The completeness measure is implemented through checklists that specify the desirable features in a research object, providing the basis for the implementation of the stability measure that checks the completeness over time.

VRC members were involved in the process to evaluate the existing checklists available in ROHUB and adapt them to Earth Science requirements. This section reports VRC requirements regarding expected checklist types and features to evaluate. In addition, the section includes a conceptual model for research objects based on checklists in Earth Science, and the implementation of this checklist in ROHUB.

4.1 Research object types and checklists in earth science

In the requirement elicitation phase CNR-ISMAR, NHP, SatCen and Supersites produced descriptions of the research object types and a list of the requirements that should be observed in time for each of them. The objective was to come up with a library of canonical checklists which a) are isomorphic to the different research object types, i.e., the purpose of the research object is defined and implemented in its compatible/assigned checklist and b) are generic enough that allow reuse by different communities even outside of the VRCs, notwithstanding customization when required.

Table 2 summarizes the research object types identified by VRCs. Most of them agreed on different research object types that mainly contain Workflows, Data, and Research Products. Two of them, CNR ISMAR and Supersites, proposed research objects focused on documentation used as support in their research. Supersites is even more specific and distinguishes research objects focussed on Bibliography, Discussion and Meetings. For each of these research object types, an associated checklist has been proposed. This deliverable focuses on research objects that are potentially publishable and cited as scholarly communications, thus the focus was on Workflow, Data and Research Product research objects.

Table 2. Research object types proposed by VRCs

CNR ISMAR	NHP	SatCen	Supersites
-	Basic	Basic	-
Workflow RO	Workflow RO	Workflow RO	Workflow RO
Data RO	Data RO	Data RO	Data RO
Research Product RO	Research Product RO	-	Research product RO
Documentation and Bibliographic RO	-	-	Bibliographic RO
-	-	-	Discussion RO
-	-	-	Meeting RO

4.1.1 Basic checklist

NHP and SatCen where the partners that identified a basic checklist so that it could be applied to all kind of research objects regardless their purpose. Table 3 summarizes the features that they propose to check in the research object and the last column called *Feature to check* defines the predicates to be included in the definitive checklist. Note that not all the proposed features are included in the basic checklist since some of these features (marked with an * in the table) are typically managed automatically by the research object management service, such as the ROHUB backend service, as the research object life cycle goes on. This is the case of research object status, creation date, version, and confidentiality. Moreover, a predicate is included to check the existence of a sketch depicting the research object internal process. This feature (in bold in the table) was not mentioned by the VRCs but in practice it has been proven very useful.

Table 3. Basic checklist.

Marked with * features that are managed by the RO management service.

NHP	SatCen	Feature to check
Title	Title	hasTitle
Description	Description	hasDescription
Owner/Creator	Owner/Creator	hasCreator
*Status	*Status	
*Confidentiality	*Confidentiality	
	*Creation Date	
	*Version	
	Tags	*hasTag*
		hasSketch

4.1.2 Checklist for workflow-centric research objects

All VRCs propose checklists for Workflow-centric research objects. The proposed checklists are summarized in Table 4 where a * indicates features which are typically managed automatically by the RO management service, such as the ROHUB backend service, and therefore they should not be tested in a checklist, a ^ indicates features out of the scope of the canonical checklist, i.e., features that are too specific for one of the VRC, not generic enough or not possible to check with a checklist.

The RO management service, such as the ROHUB backend service, manages features as creation date, modification date, RO type, source RO, status, version, confidentiality, and quality. Features that were considered non-valid since they were not applicable for the checklist or not generic enough are: workflow included in a research paper, and workflow in Taverna format. The rationale is that it is not possible to verify in a checklist that an academic paper includes a workflow, and the research objects are not tied to Taverna.

The features that were considered out of the scope of the canonical checklist are *approved by*, *checked by*, and *funding*. These features are part of a governance and financial model that are specific to each VRC. This decision does not preclude the usage of these features in more specific checklists in the future, and therefore the RO model should be extended in order to specify the appropriate annotation properties.

Additionally, the copyright owner feature (in cursive in the table) is currently not supported in the research object model and therefore, it was extended to specify the appropriate annotation property. Finally, the *hasWorkflowRun* predicate was added so that workflow-centric research objects can test the existence of execution provenance information.

Table 4. Checklist for workflow centric research objects.

Features with * are managed by the RO management service, with – are non-valid features, and with ^ are not covered by the canonical checklist

CNR ISMAR	NHP	SatCen	Supersites	Feature to check
Title	Title	Title	Title	hasTitle
Description	Description	Description	Description	hasDescription
Owner/Creator	Owner/Creator	Owner/Creator	Authors	hasCreator
			^Affiliations	
		*Date	*Created on	
			*Last modified on	
*Type			*RO Type	
			*Source RO	
Workflow is present	Workflow present?	Workflow present?	**^**Workflow in Taverna format	hasWorkflow
Copyright Owner	Copyright Owner	Copyright Owner		*hasCopyrightHolder*
*Status	*Status	*Status	*Status	
		*Version		
	*Confidentiality	*Confidentiality		
	Data/Data Link present?	Data/Data Link (if present)		hasInputData
	Data Description - format	Data Description - format		dataHasFormat
	Data Description - size	Data description - size		dataHasFileSize
	Output present?	Output (if present)		hasOutputData
	*RO Quality	*RO Quality		
Wf definition				hasWorkflowDefinition
web serv. accessible				areWebServiceAccesible
soft. dependencies				hasRequirements
^wf. in paper				
RO design sketch				hasSketch
		Ancillary documents		isDocumentedBy
	^Checked by	^Checked by		
	^Approved by	^Approved by		
			^Funding	
		Tags		hasKeywords
				hasWorkflowRun

4.1.3 Checklist for data-centric research objects

For research objects focused on data, VRCs propose the checklists presented in Table 5. This table has filtered out all the features that were considered invalid or out of the canonical checklists scope as in the previous cases of workflow-centric research objects and the basic checklist. Within the remaining features, the Editor (as a journal editor) is not supported by the research model and therefore it was extended with this new term. Additionally, as described for the workflow checklists, the research object model was extended to support testing the predicate *hasCopyrightOwner*.

Table 5. Checklists for data-centric research objects.

CNR ISMAR	NHP	SatCen	Supersites	Features to check
Title	Title	Title	Title	hasTitle
Description	Description	Description	Description	hasDescription
Owner/Creator	Owner/Creator	Owner/Creator	Authors	hasCreator
Purpose	Purpose	Purpose		hasPurpose
	Editor	Editor		*hasEditor*
Copyright Owner	Copyright Owner			*hasCopyrightOwner*
	Data Description - format	Data Description - format		dataHasFormat
	Data Description - size	Data description - size		dataHasFileSize
			Data	hasInputData
DOI present				hasDOI
Access level				hasAccessLevel

4.1.4 Checklist for Research Objects describing research products

Regarding checklists for research objects modelling research products CNR ISMAR, NHP and Supersites provided feedback. Table 6 shows the features proposed to test in these checklists. With the exception of *hasCopyrightOwner* and *hasOutputData* that were previously identified as missing in the current research object model all the features are included in the model and therefore their corresponding predicates can be evaluated in the checklist.

Table 6. Checklist for research objects describing research products

CNR ISMAR	NHP	Supersites	EVER-EST
Title	Title	Title	hasTitle
Description	Description	Description	hasDescription
Owner/Creator	Owner/Author/Co-author	Authors	hasCreator
	Contributor		hasContributor
Purpose/Use of Data			hasPurpose
		Workflow/SW code/ interpretation process	hasProcessImplementation
Copyright Owner			*hasCopyrightOwner*
Access level			hasAccessLevel
	References		hasReferences
	Data link	input	hasInputData

CNR ISMAR	NHP	Supersites	EVER-EST
		output	hasOutputData
	DOI		hasDOI

4.2 EVER-EST research object types taxonomy

The analysis of the requirements produced by the VRCs in terms of research object types and their associated checklists resulted in the design of a taxonomy where the different types of research objects can be classified (see Table 6). This taxonomy has the characteristic that a research object of a particular type can be validated against the corresponding checklist specified for its type or against any of the checklists of its parent types. Two main types of research objects were distinguished, those for investigations focused on the processes, and those for investigations focused mainly on data artefacts. Research objects focussed on processes can be specialized in research objects containing programming code, web services and workflows. For each of these research types a canonical checklist can be associated.

Note that in addition to the checklist applied to the research object type, e.g., workflow research object, the checklist for process research object (not discussed here) and the basic checklist also apply. Similarly, the data-centric research object has as specialization the research product research object. It is expected that a research product research object contains or references data artefacts, or references another research object containing the data. Such condition is specified in the definition of the data-centric research object, and therefore, applies to the research product research object. Additionally, it is expected that a research product, e.g., a scientific publication, should link the computational process used to achieve the published results. This condition is specified in the definition of the research product research object to make sure it contains/reference some process artefact.

The checklists associated to these RO types are going to be available at: https://github.com/wf4ever/ro/tree/earth-science/checklists

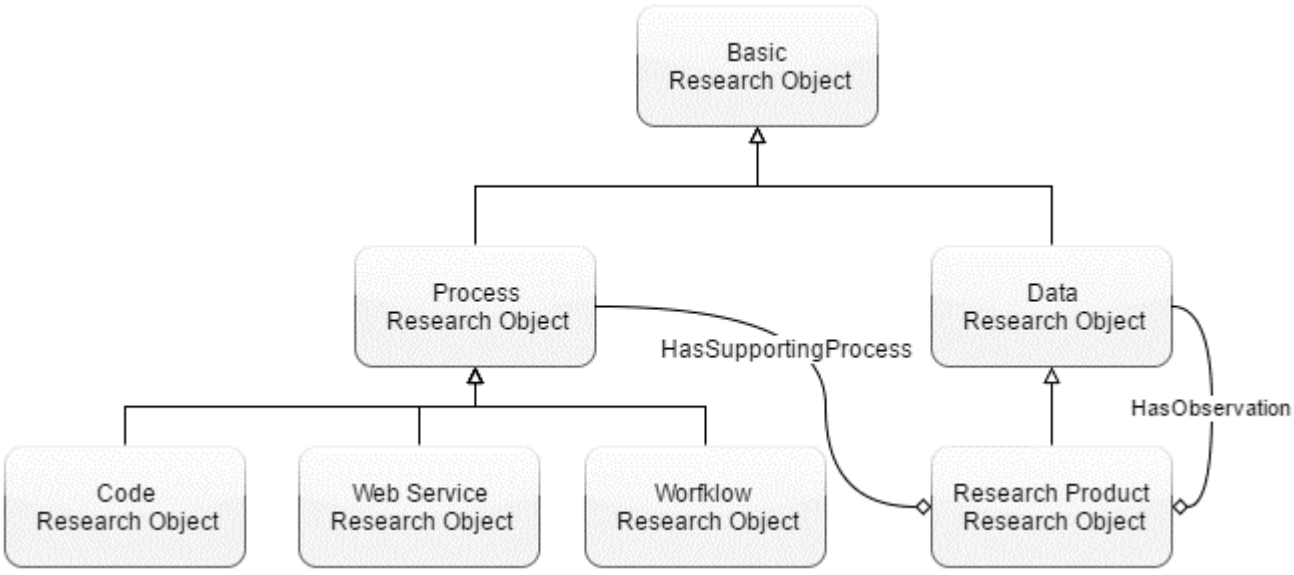

Figure 16. Research Object type taxonomy for earth observation.

4.3 Checklists in support of the data management plan

The EVER-EST data management plan has been defined around the FAIR principles that state that data must be Findable, Accessible, Interoperable and Reusable (FAIR) in addition to specific requirements from the VRCs about data privacy, intellectual property, and long term preservation. Accordingly, research objects follow the same FAIR principles, aiming at supporting the adoption of better practices in Earth Science disciplines.

The research object model and related technologies enable findable, accessible, interoperable and reusable data through the use of common ontologies to describe the metadata, a W3C standard data format specifically designed for data interoperability on the Web, and the use of linked data technology as publication mechanism.

In addition, the data management plan includes data quality assurance processes as a way to support long-term preservation and reuse. Therefore, the checklists presented above are an important tool to support the implementation of the EVER-EST data management plan.

4.4 Summary of changes in the RO model

The previous analysis also gave rise to a few updates in the latest version of the research model ontologies and vocabularies reported in D4.2. In particular, the following changes have been identified and implemented for the Earth-Science extension (roes):

- Rename class roes:MinutesResearchObject as roes:DiscussionResearchObject (the requested MeetingResearchObject can be considered as a snapshot of the DiscussionResearchObject).
- Add class roes:CodeResearchObject
- Add class roes:ResearchProductResearchObject
- Add class roes:ProcessResearchObject
- Add in roes:ProcessResearchObject subclass of ro:ResearchObject
- Add in roes:WebServiceResearchObject subclassOf roes:ProcessResearchObject
- Add in wf4ever:WorkflowResearchObject subclassOf roes:ProcessResearchObject
- Add in roes:CodeResearchObject subclass of roes:ProcessResearchObject
- Add in roes:ResearchProductResearchObject subclass of roes:DataResearchObject
- Add objectProperty voag:isApprovedBy
- Add objectProperty pav:curatedBy
- Add objectProperty foaf:fundedBy
- Add objectProperty roes:editedBy
- Add objectProperty odrs:copyrightHolder

The following namespaces are being used:

- ro=http://purl.org/wf4ever/ro#
- roes=http://w3id.org/ro/earth-science#
- wf4ever=http://purl.org/wf4ever/wf4ever#
- voag= http://voag.linkedmodel.org/schema/voag#
- pav=http://purl.org/pav
- foaf=http://xmlns.com/foaf/0.1
- odrs=http://schema.theodi.org/odrs#

4.5 Technological support in ROHUB

As part of the VRE portal, ROHUB enables Earth Science users to monitor research object integrity and authenticity by integrating a set of functionalities that implement quality metrics such as completeness, stability and reliability.

4.5.1 Checklist Evaluation

Completeness is calculated in the research object model with the help of checklists. ROHUB currently enables users to assess the quality of a research object against any of the available checklists. In order to perform this task, the user must open the research object, and from the research object quality tab, select the desired checklist (see Figure 17Figure 1). The results are then displayed with green checks the features that are successfully tested in the research object, in yellow crosses the features that are not successfully tested but are not compulsory in the checklist, and in red crosses the features that are not successfully tested and are compulsory. For each feature, a textual description of the test is provided. Additionally, from this tab, the user is able to open the research object monitoring tool, an external application, which presents the quality of the research object throughout time in terms of its completeness, stability and ultimately reliability.

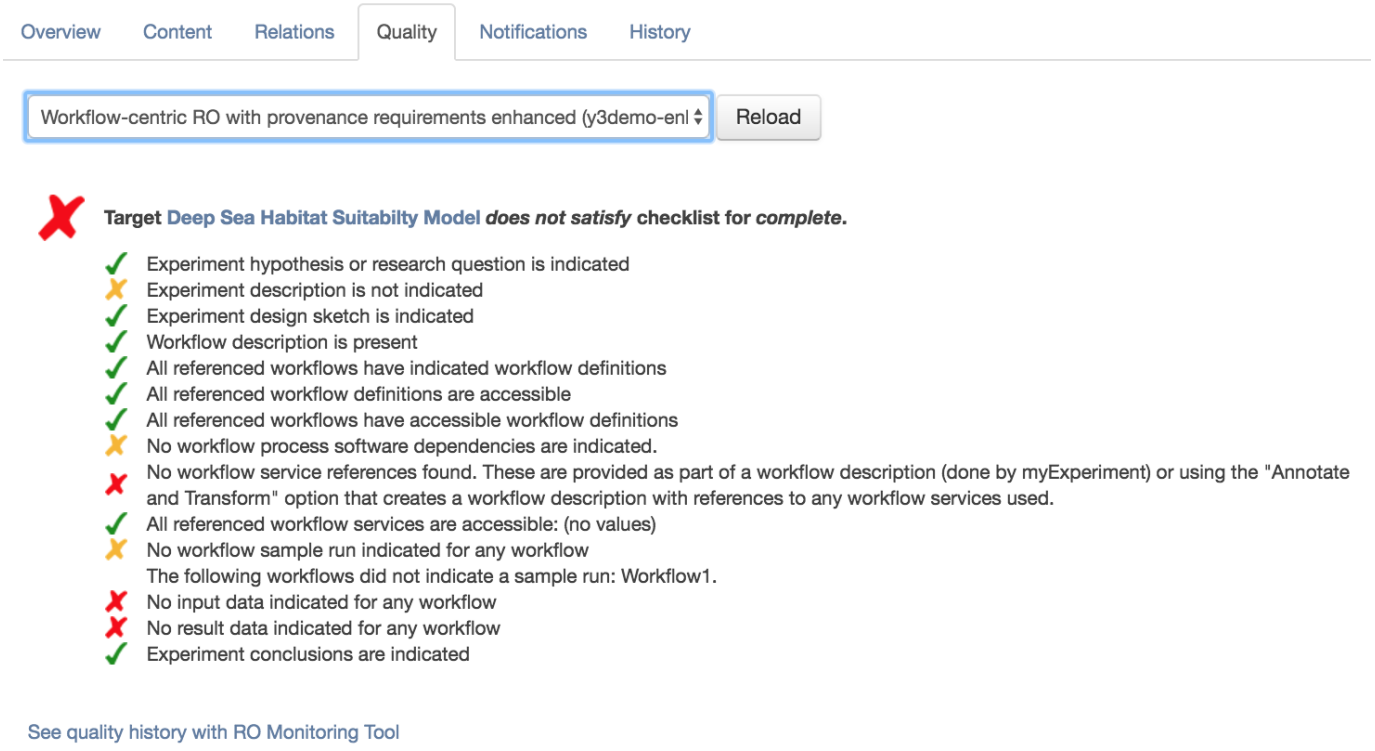

Figure 17. checklist evaluation in ROHUB

In addition, ROHUB displays a basic indicator of the RO quality in the RO overview tab (Figure 18, Figure 17), which represents the compliance of the research object against the most basic checklist. The indicator is in the form of a completeness bar, and when the user clicks on it, a detailed description of the checklist evaluation tests is displayed.

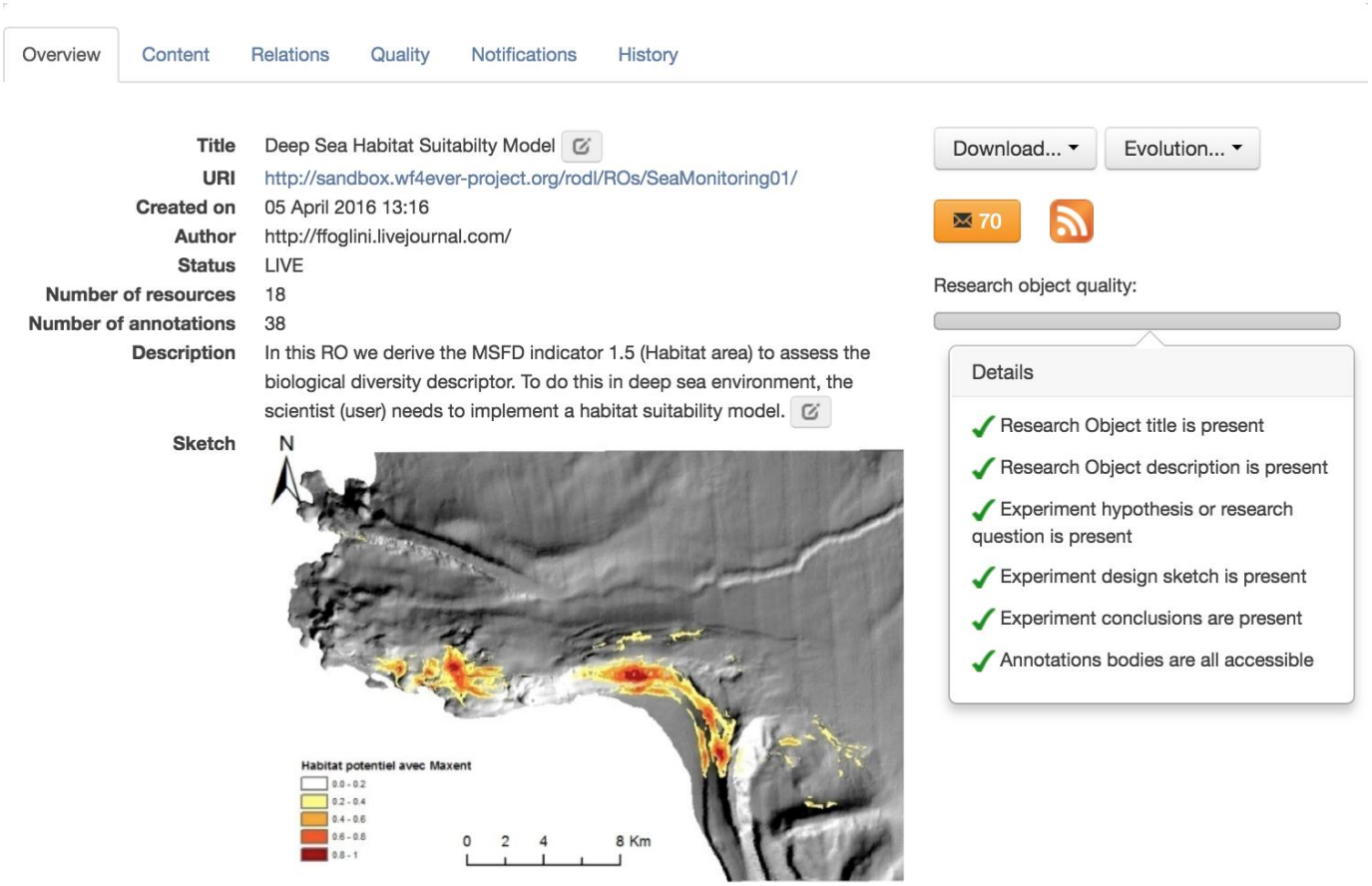

Figure 18. Basic RO quality evaluation in ROHUB

4.5.2 Research object monitoring

Stability and reliability metrics, introduced in D4.1, aim to keep track and measure the changes of the completeness assessment of a RO throughout time. The stability measures the ability of a RO to maintain its status during its lifetime while reliability extends stability to incorporate the completeness assessment. These measures have been specified in detail in [8]. ROHUB portal integrates the RO monitoring tool that support the implementation of these measures. The Ro monitoring tool computes completeness, stability, and reliability scores in a time interval and saves the results as additional metadata within the research object. The tool interface visualize these metrics as a chart (see Figure 19) allowing a more comprehensive view of the quality information of a research object. More details about this tool usage and installation can be found in deliverable D5.3.

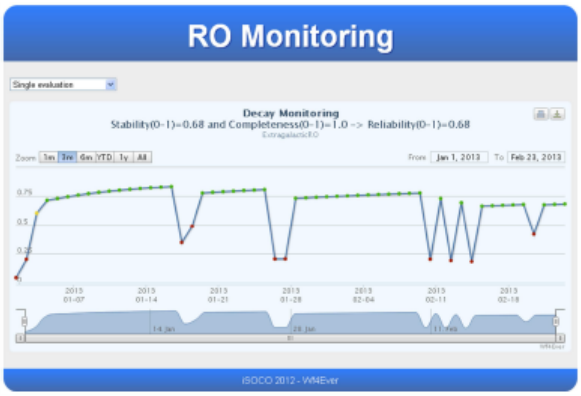

Figure 19. Research object monitoring tool

5 Components for Enabling Research Objects as Scholarly Communications

Traditional scholarly communications in their digital version do not facilitate, to the extent that is required by scientific communities, the exchange and reuse of research data, methods and results. In this context, the research object model along with similar initiatives such as ORE OA[4], which propose containers of digital resources that contextualize a research publication, have emerged as new paradigm[5]. The research object model is digital native since it was conceived as an aggregator of digital content that could be easily interchanged on the web. Research objects provide metadata readable by researchers and software agents that can use it for data exchange, visualization, reuse and preservation. However, the model needs a mechanism to ensure proper credit to research objects when they have been reused in other research objects. Therefore, the research object model has been enhanced with Digital Object Identifiers DOIs, which are the de facto standard mechanism in scholarly communications, ensuring appropriate credit to research outcomes.

5.1 Digital Objects Identifiers (DOIs) for research objects

The DOI system provides an infrastructure for persistent unique identification of objects of any type. It was designed and implemented by International DOI foundation IDF and has been standardized through the International Standard Organization in 2012 (ISO 26324). The DOI system was designed for network awareness and interoperability, which makes it a good complement to the research object model. The use of DOIs allow to calculate research object impacts metrics, since systems can track the publications in scholar tracking systems such as Google scholar and Microsoft academics.

DOIs are persistent identifiers that are used to unique identify an object, either physical or digital. A **DOI name** may be assigned to any entity, which must be precisely described with structured metadata. The DOI name has two components, a prefix and a suffix that are separated by the slash character (/). The prefix refers to a unique naming authority, and the suffix is an identifier chosen by the registrant. This combination avoids the necessity of a centralized allocation of DOI names. An example of DOI name is 10.1594/PANGAEA.726855 where 10.1594 is the registration agency identifier, DataCite in this case, and PANGAEA.726855 is the object identifier assigned by DataCite. In addition to names, the DOI system comprises a **name resolution service** that redirects a name to associated data. DOI names can be resolved using the IDF name resolver: http://dx.doi.org/

The DOI system documentation provided by the IDF is comprehensive. The reader is referred to the DOI handbook for details about the DOI system[6]. In this section, the focus is on the integration of DOIs in the research object lifecycle and the requirements that registration agencies pose to assign DOIs to research objects.

5.2 Research Object Lifecycle

The lifecycle refers to the stages that the RO transitions from its conception until its conclusion. In the wf4Ever project different lifecycles scenarios were analysed (see Figure 20) and the possible research objects states where identified [7]:

- **Live ROs**: represent a work in progress. They are thus mutable as the content or state of their resources may change.
- **Snapshot ROs**: are intended as a record of past activity, ready to be disseminated as a whole. They are immutable, and reflect the state of the Live RO at a certain time.

[4] https://www.openarchives.org/ore/

[5] https://www.w3.org/community/rosc/

[6] https://www.doi.org/hb.html

- **Archived ROs**: represent the final stage of a RO where it has either reached a version that the author prescribes to be stable and meaningful and is appropriate for publication and long-term preservation, or it has been deprecated. They are therefore immutable, with no further changes or versions allowed.

The challenge was to design a lifecycle scenario to support the DOI assignation to research objects that are ready for publication. The result of this design is the scenario depicted in Figure 21. The scenario starts when the researcher creates a research object. The research object management platform will automatically assign the Live state unless the author specified it otherwise. Once the earth scientist considers that the research object has reached some milestone and its content is worth of publication or sharing, for instance as a technical report for internal distribution or as scholar contribution to the research community, he can proceed to generate a snapshot. As mentioned before a snapshot is a copy in a point in time of a Live research object, and as such they are immutable. Thus, research object snapshots are good candidates to receive a DOI since, according to the definition, they are ready to be shared and will not change. It will be up to the scientist to decide if they want to assign a DOI when they generate a research object snapshot, and up to the research object management platform (e.g., ROHUB) to verify that the research object fulfils the requirements, drawn from the research object checklist, to get it.

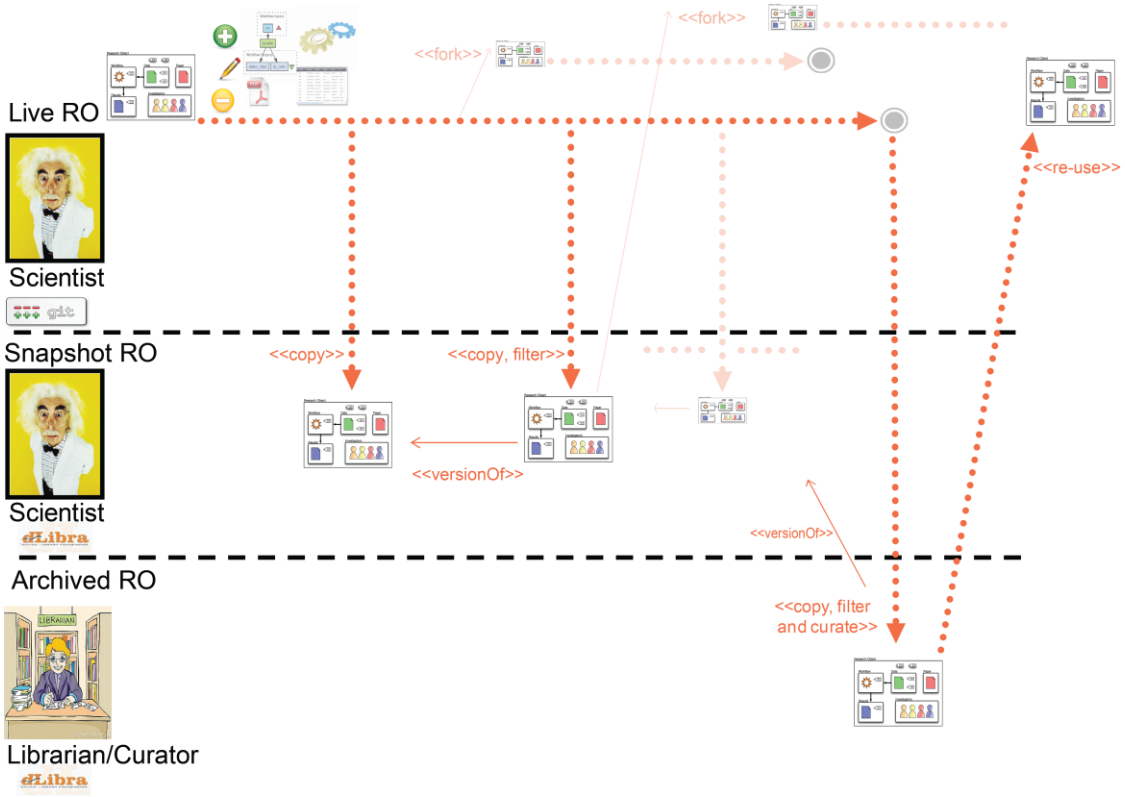

Figure 20 Research Object Lifecycle Scenarios [7]

When the researcher considers investigation to be final the research object is archived. This is the state for research objects that have reached a maturity degree such that its content is not going to be modified in the near future, or that have been deprecated (e.g., research line stopped or found useless). Note that in a typical scenario, a Live research object gives rise to a number of snapshot research objects before producing an Archived research object.

Moreover, throughout the research object lifecycle, new lines of work can be started at any point in time (e.g., to explore different hypothesis, test different configurations), either while the research object is alive by forking the

Live RO at some point in time, or once the RO has reached a milestone/stable state by forking any of the RO snapshots or the RO archive. In short, a new Live research object, either from scratch (as in Figure 21) from a set of existing resources or by performing a fork operation, can give rise to multiple snapshots (after reaching different milestones), and when the investigation ends, an archived research object is generated.

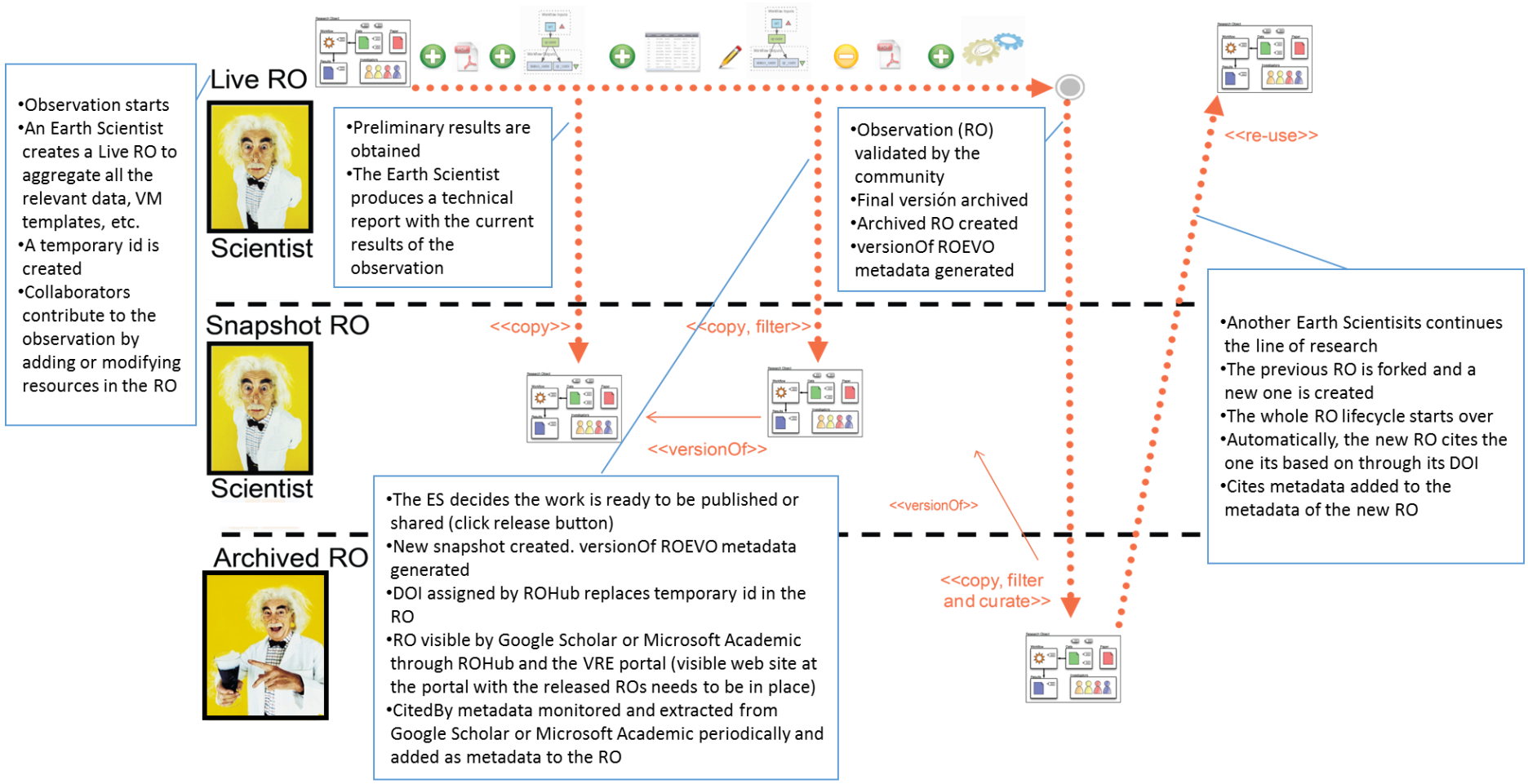

Figure 21 Research Object lifecycle scenario supporting DOI generation.

5.2.1 DOI generation

The norm ISO 26324 named the International DOI foundation as the ISO 26324 registration authority, and it defines a controlled namespace for DOIs that is populated by the registration authority. This means that anyone that wants to generate DOIs have to be authorized by the IDF. Registration Agencies implements the DOI system by providing domain-specific identifiers for various applications using the DOI framework. For instance, Crossref[7] provides DOIs for scientific publishing, and DataCite[8] generates DOIs for referencing and sharing scientific datasets.

Therefore, the first task was to identify a DOI provider. The British library[9], which is a member of DataCite, showed interest to be the DOI provider for research objects. The British library basic requirements for organizations that wants to assign DOIs are: i) authority to assign DOIs to data, ii) guarantee data persistence, iii) data is accessible to external users, and iv) data has citation potential. The British Library agent agreed with the fact the ROHUB digital library and the research objects fulfil all the aforementioned requirements.

One of the main concerns was about the future of the ROHUB platform after the EVER-EST project end. The Poznan Supercomputing and Networking Centre PSNC as ROHUB host, through a letter expressed their interest to maintain the ROHUB digital library after the end of EVER-EST. Other requirements of the British Library were the following responsibilities that PNSC agreed to satisfy:

- Provide and maintain at least the mandatory DataCite metadata for each item with a DOI.
- Make metadata openly available without restriction (under Creative Commons Zero waiver).
- Maintain publicly accessible landing pages for each item with a DOI.

5.3 Research object impact

DOIs are used to track research objects in scholarly search services so that citation information can be extracted. Based on the citations, reliable impact metrics are calculated that can be included as research object metadata. The goal is to harvest citation information from Google Scholar and Microsoft Academic. Google Scholar and Microsoft Academic are specialized sites that gather citation information of research works in scholarly communications. Note that most of the research objects in ROHUB do not have a DOI since this is a new functionality, and therefore the current release of the scholar services use the research object title to gather citation information from Google Scholar and Microsoft Academics.

Google scholar does not provide an application programming interface API to access to its content. Therefore, the only way to extract the web site information regarding publications is by scrapping the web pages in regular time intervals. Thus a RESTful web service has been developed to look for research objects in google scholar by using the title. This service scraps the citation information and has been implemented on top of scholar.py[10], a python module that implements a query system and parser for Google Scholar's output. The service receives the title of the publication as a path parameter and is deployed at:

http://everest.expertsystemlab.com/scholar/gscholars/write-title-here

On the other hand, Microsoft Academic provides a RESTful API, called Academic Knowledge API[11], which allows querying the scholarly information. This API was used in a java program to retrieve regularly citation information about research objects, once again using the research object title. The next release of the research object

[7] http://crossref.org/

[8] https://www.datacite.org/index.html

[9] http://www.bl.uk/aboutus/stratpolprog/digi/datasets/datacitefaq/faqhome.html

[10] https://github.com/ckreibich/scholar.py

[11] https://www.microsoft.com/cognitive-services/en-us/academic-knowledge-api

components (deliverable D4.4) aims at integrating such citation information as part of the research object model so that research object authors, contributors, and other interested members of the community can assess the impact of their research contributions through the research object metadata. The deployed service receives the publication title as a path parameter:

http://everest.expertsystemlab.com/scholar/msacademics/write-title-here

5.4 Technological support in ROHUB

The technological support for the lifecycle management of research objects in the context of the VRE is provided by ROHUB, which is also now being extended to support the assignment of DOIs. In particular, ROHUB provides a set of visual components enabling users to manage the research object evolution (5.4.1).

5.4.1 ROHUB portal

Currently, ROHUB enables to create RO snapshots or RO archives from a Live RO. For this task, users open a Live RO in the Portal and from the overview tab select the snapshot or release (archive) button in the Evolution dropdown list, as depicted in Figure 22. The snapshot or archive is then created in a background process, which performs the following operations:

- Performs the two operations of the API in one step, i.e., the creation of the RO copy and the finalization of the state transformation (as described above).
- The new RO is named automatically as RO-id-snapshot[-x], where is x is an incremental number starting from 1.
- Generates evolution annotations according to the RO evolution model:
  - The reference to the source research object from which this snapshot/archive was generated
  - The date/time of the snapshot/archive generation
  - The person that generated the snapshot/archive
  - The reference to the previous snapshot in the RO lifecycle
- Calculates the changes of the newly generated snapshot/archive with respect to the previous snapshot in the RO lifecycle, and represent them according to the RO evolution model.

The current interface will be improved and extended in ROHUB portal v2, in order to enable users to launch a wizard providing the following options before finalising the state transformation:

- Specify the name of the snapshot/archive.
- Modify the temporal copy of the RO, e.g., to perform some curation tasks.
- Generate additional annotations:
  - Assign version annotation property automatically
  - Citation of the source research object automatically
- Possibility to assign a DOI for the snapshot/archive.

Not that in the case the user wants to assign a DOI to the snapshot/archive, a checklist will be evaluated against the research object to make sure that it complies with the minimum requirements to obtain such identifier.

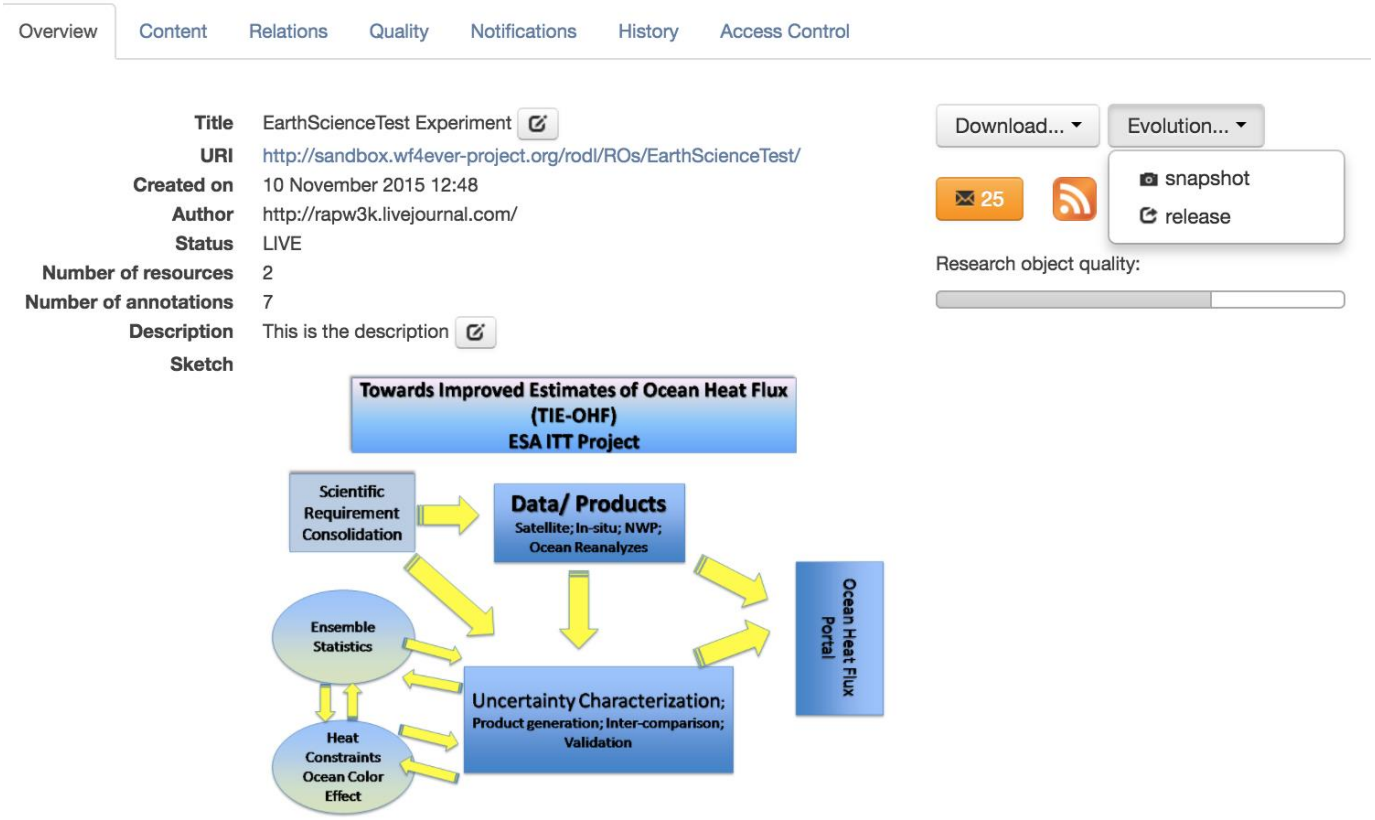

Figure 22. Interface for creating RO Snapshot or RO Archive in ROHUB portal

Additionally, the current ROHUB portal enables users to inspect the evolution of a research object from the RO History tab, as depicted in Figure 23. This tab shows a graph of ROs or similar resources that are related to the inspected RO in terms of its evolution. If the current RO is a snapshot, the graph shows its live RO and previous snapshots, for example. The graph is built using the information returned by the RO evolution API. The user can navigate throughout the different states by clicking on any of the displayed objects in order to open it. Although at the moment snapshot and archive ROs can be created only from Live ROs, in the graph not all resources must be ROs in a strict sense, for example the Live RO may be a URI pointing to a stack of files with no metadata.

In ROHUB portal v2, this interface will be improved/extended.

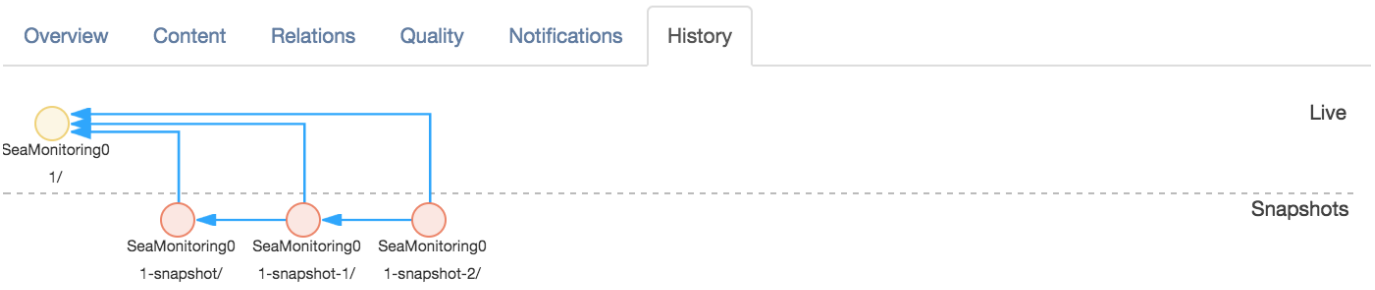

Figure 23. RO evolution history in ROHUB portal

6 Conclusions and Future Work

This document presented the main research object components that support the preservation of scientific knowledge and enable its sharing and reuse. These components include search and recommendation tools, checklist to assess the quality of research objects, the use of DOIs to identify research of objects as scholarly publications, and mechanism for keep track of research object impact in scholarly communications.

The search and recommendation tools support Earth Scientists by offering advanced search functionalities in the case when they know exactly what they are looking for, and recommending research objects when the goal of the search is fuzzy. The search engine has a remarkable feature; it understands the semantics of the research object content with the help of a semantic network where terms are mapped to concepts which are related among them. Thus the search engine is able to identify, in the research object, concepts, domains, and named entities, including people, location and organizations. This semantic search engine improves the current search system that lacks of semantics and only uses the research object user-generated metadata. The recommender uses an intuitive interface based on the Collaboration Spheres concept that eases the definition of the recommendation context and result presentation. The recommender also includes semantic processing of the research object content but it takes the semantic processing one step further by including word embeddings representations, as a complement to the explicit semantic approach based on semantic networks. Word embeddings allows to rapidly cover domain specific terminology that is not contained in the semantic network.

Regarding research object quality assessment, checklists were designed following the requirements of VRC members. VRCs identified the following checklists: basic, workflow, data, and research product. These checklists were analysed to decide which features were actually to be included in the checklists. As a result the research object model was modified to support new terms that were not covered by the original model. The checklists give way to a taxonomy of research objects where each checklist correspond to a research object type. Hence, the quality of a research objects can be assessed with the corresponding checklist to its type of with the checklists of the parent types in the taxonomy.

Digital object identifiers were incorporated in the research object model and lifecycle to integrate research objects in the scholar communication context. DOIs enhance research object visibility since they are automatically indexed by prominent scholarly sites, allow proper credit to research object authors when their work is reused, and ease the collection of citation information from scholarly tracking sites. In this respect two services were deployed that allow collecting citation information of research objects in Google Scholar and Microsoft Academics.

Future work planned for next deliverable D4.4 includes the improvement and maintenance of the components presented in this deliverable and their integration in ROHUB and the virtual research environment. The plan is to have a single search engine that integrates the semantic search engine and the current search system deployed in ROHUB. The recommender system will be integrated in ROHUB and the semantic processing will be enhanced with word embeddings. In addition, checklists for research objects compiling information about bibliographies, discussions and meetings will be analysed and implemented in ROHUB. Finally, ROHUB will include the research object citation information that has been collected from scholarly citation tracking sites.