

Article

Describing Geospatial Assets in the Web of Data: A Metadata Management Scenario

Cristiano Fugazza ^{1,*}, Monica Pepe ¹, Alessandro Oggioni ¹, Paolo Tagliolato ^{1,2,3}, Fabio Pavesi ¹ and Paola Carrara ¹

¹ Institute for Electromagnetic Sensing of the Environment, National Research Council (IREA-CNR), v. Bassini 15, Milan 20133, Italy; pepe.m@irea.cnr.it (M.P.); oggioni.a@irea.cnr.it (A.O.); tagliolato.p@irea.cnr.it (P.T.); pavesi.f@irea.cnr.it (F.P.); carrara.p@irea.cnr.it (P.C.)

² Institute of Marine Science, National Research Council (ISMAR-CNR), Tesa 104—Arsenale, Castello 2737/F, Venice 30122, Italy

³ LifeWatch Italy, 73100 Lecce, Italy

* Correspondence: fugazza.c@irea.cnr.it; Tel.: +39-02-2369-9456

Academic Editors: Michael Leitner and Wolfgang Kainz

Received: 25 July 2016; Accepted: 29 November 2016; Published: 2 December 2016

Abstract: Metadata management is an essential enabling factor for geospatial assets because discovery, retrieval, and actual usage of the latter are tightly bound to the quality of these descriptions. Unfortunately, the multi-faceted landscape of metadata formats, requirements, and conventions makes it difficult to identify editing tools that can be easily tailored to the specificities of a given project, workgroup, and Community of Practice. Our solution is a template-driven metadata editing tool that can be customised to any XML-based schema. Its output is constituted by standards-compliant metadata records that also have a semantics-aware counterpart eliciting novel exploitation techniques. Moreover, external data sources can easily be plugged in to provide autocompletion functionalities on the basis of the data structures made available on the Web of Data. Beside presenting the essentials on customisation of the editor by means of two use cases, we extend the methodology to the whole life cycle of geospatial metadata. We demonstrate the novel capabilities enabled by RDF-based metadata representation with respect to traditional metadata management in the geospatial domain.

Keywords: spatial data infrastructures; geospatial metadata; RDF; semantics; editor customisation

1. Introduction

Retrieval of assets on the Web primarily relies on the metadata describing them. Data portals can then articulate search mechanisms on the basis of these descriptions, with degrees of expressiveness that are dependent on the granularity of the metadata schema governing their structure. Among assets, those related to geospatial information are relying almost entirely on metadata for retrieval, often dubbed as *discovery*, because assets are typically in non-textual format and then the indexing practices of generalist search engines are inefficient at best. Moreover, this category of data is characterised by properties (e.g., the geographic extent, a.k.a. the *bounding box*) that require standard metadata in order to be encoded and processed by the specific-purpose tools in this domain [1–3]. This is among the drivers of the standardisation effort represented by the INSPIRE Directive, [4–6], a collection of recipes for interoperability that have sound foundation on standards. INSPIRE allows for harmonisation of geospatial assets among European countries because, despite the different transpositions of the Directive in the distinct countries, the baseline set by core INSPIRE metadata allows the geospatial community to discover and access heterogeneous assets in a seamless fashion.

However, the landscape of data management has inevitably changed since the formulation of the Directive and its early, prototypical implementations. With regard to metadata, the once-state-of-the-art

groundwork set by the reference standards for geographic information [7–9] has become inadequate, primarily because of the shift in perspective from data representation (that is, the quest for appropriate encoding mechanisms for data and metadata in a specific domain) to data access and processing. As an example, terms like Open Data, Linked Data, RDF, and SPARQL are increasingly associated with data interoperability and open-ness. Also, novel requirements not envisaged in the original framework of INSPIRE have emerged in the last few years, some of these are being tackled by INSPIRE Thematic Working Groups (such as in [10,11]). As an example, data catalogues—in the broadest sense—have been enriched by the new category of data sources constituted by real-time/near-real-time data pulled from sensors. Consequently, these new data sources required ad-hoc data and metadata representations for management and enactment. Finally, as formats and practices develop in the geospatial context, the Web at large participates in the emergence of the vast amount of machine-processable information generally referred to as the Web of Data [12,13]. In general, Semantic Web technologies have a great potential for fine-grained discovery of assets, as testified by the applications of semantics in many distinct domains, such as in [14–16].

In the engineering of the data sharing infrastructure for RITMARE (<http://www.ritmare.it/>) (a Flagship Project by the Italian *Ministero dell'Istruzione, dell'Università e della Ricerca*) we decided to adopt best practices that, together with compliance with the INSPIRE Directive, could also ease migration to the aforementioned novel paradigms for data representation and access. In fact, on the one hand we need to harmonise metadata according to the baseline set by INSPIRE (specifically, by the Italian transposition of the Directive, RNDT [17]) and also according to SensorML, the metadata format we employ for sensor information [18,19]. On the other, we want to ground metadata creation on controlled vocabularies and, in general, context information drawn from the Web of Data. Referring to the categorisation of heterogeneities in [20], our research tackles the semantic level, the other two (syntactic and structural) being addressed by the authoritative standardisation bodies (e.g., the ISO, INSPIRE, and OGC communities). Also, it should be noted that, among the different efforts addressing semantic mismatch of metadata descriptions, we are concentrating on semantic lift (that is, the association of text-based property values with unique, URI-based identifiers, as detailed in Section 3). Other research threads, such as the complementary activities devoted to harmonising the independent contributions to the geospatial Web of Data into a consistent mesh of interconnected data structures, shall be considered as off topic.

In other words, we present our solution for bridging the gap between legislative compliance and the aforementioned novel Web 3.0 practices [21]. In particular, we focus on the integration of external data sources in order to turn single-tenanted, monolithic metadata descriptions into *living documents*. In fact, our RDF-based metadata descriptions can more easily adapt to changes (researchers changing workplace, companies moving to a different address, terminologies evolving, etc.) w.r.t. traditional, text-based metadata and elicit innovative discovery mechanisms, as exemplified in Section 4. It should be noted that, whereas the practices described in this work stem from the requirements posed by the RITMARE project, usage of the methodology does not require any prior knowledge on the project. In fact, our metadata management framework has been adopted by a number of projects, namely the FP7 projects ERMES (<http://www.ermes-fp7space.eu/it/homepage/>) and EuroFleets2 (<http://www.eurofleets.eu/np4/home.html>), the H2020 project eLTER (<http://www.lter-europe.net/lter-europe/projects/eLTER>), and the Italian Flagship Project NextData (<http://www.nextdatapoint.it/>). Moreover, the metadata profiles that are available out-of-the-box are of general interest to the INSPIRE and SWE communities [22] and then our methodology is fit for adoption by data providers.

The methodology for metadata management we propose is hinged on *templates*; that is, definitions of metadata schemas created according to a metalanguage (here, we use “metalanguage” instead of “template language” to avoid ambiguity with the definitions in Section 3.2). Templates express the structure of a target XML metadata schema, such as the already mentioned INSPIRE and RNDT schemas, but also SensorML (versions 1.0.1 and 2.0.0) [18,19]. They also contain the necessary information to apply semantic lift to the metadata that is generated. Specifically, the template drives

creation of a metadata editing interface, the application frontend, that is bound to the intended schema and data sources. Then, the metadata authoring activity provides additional information that is not meant to the specific XML output but rather to constituting the semantic counterpart to the metadata description, expressed as RDF [23]. Whenever the original metadata are requested, for example after discovery of the asset by the end user, the template is looked up again, the RDF description corresponding to the asset is extracted from the catalogue, and the XML description is reconstructed. The advantages of this practice with respect to metadata consistency are apparent: In fact, every time the XML metadata description is generated, it is built on the basis of the updated information drawn from the Web of Data according to what the template and the RDF description specify.

This paper is organized as follows. Section 2 draws the context of this work, comprising essentials on Semantic Web technologies that are going to be useful in the following, presents an overview of project RITMARE, and introduces related works. Among the solutions proposed, we focus on EDI, the metadata management application harnessing XML and RDF for metadata description. Section 3 presents two use cases and implements them as template structures; this Section also covers integration of external data structures and generation of the distinct output formats. Section 4 discusses the advantages of our approach. Finally, Section 5 draws conclusions, hints at the “big picture” of decentralised metadata management, and outlines future work on the subject.

2. Context

2.1. Semantic Web Essentials

Among the innovative aspects of our approach is recourse to information drawn from the Web of Data for metadata provision. We are going to barely scratch the surface of the variety of technologies, specifications, and practices that characterise this domain but, after reading this paragraph, the reader shall be aware of the data representation and query formalisms that are used throughout the paper. Further information on this topic can be found in [24]. Basic knowledge of the XML data model (<https://www.w3.org/standards/xml/>) is also useful to fully grasp the structure of the metalanguage we developed; still, introducing this baseline technology is out of the scope of this work.

The reference data model for the Web of Data is RDF [23], a formalism expressing information as directed labelled graphs whose atomic component is the *triple* (or *assertion*), such as the following:

```
<http://some/subject> <http://some/predicate> <http://some/object> .
```

Triples are composed of a *subject*, a *predicate*, and an *object* typically identified through URIs [25] (think of URIs as good old URLs that do not necessarily lead to some web content). *Objects* can also be plain literals, although this category of “leaf” entities does not contribute to shaping the overall data graph:

```
<http://some/subject> <http://some/predicate> "some object"@en .
```

The *namespaces* [26] hampering readability of URIs are often substituted with prefixes, as in the Turtle formalism we chose among the different serialisation formats for this data model [27]. Turtle also allows to avoid specifying invariant information (i.e., same property and/or same subject) when providing multiple triples by using punctuation:

```
@prefix ex: <http://some/> .
```

```
ex:subject    ex:predicate_1  ex:object_1  ;
              ex:predicate_2  ex:object_2  .
```

```
ex:subject    ex:predicate_3  ex:object_3  ,
              ex:object_4  .
```

Whatever the specification style, RDF triples induce a (decentralised) graph whose structure and information content require ad-hoc query and, in general, manipulation formalisms. In an ideal Linked

Data scenario (that is, when URIs in RDF descriptions are resolvable to the actual data structures), agents can cross the graph from one end to the other by following RDF properties; in realistic terms, browsing the Web of Data requires federating a number of distinct *endpoints* that are interrogated through SPARQL, an SQL-like query language.

SPARQL [28], has reached maturity and provides far more flexibility and expressiveness than traditional SQL [29] (albeit the performance that relational databases allow for when managing huge amounts of data is still unparalleled). We are only going to use SPARQL queries featuring, in accordance with SQL practices, SELECT and INSERT statements, even if the language allows for other query forms (CONSTRUCT, DESCRIBE, ASK). Retrieval of data with SELECT statements amounts to matching the data (i.e., the triples, the assertions) in the data base with the graph defined by the triples in the query. As an example, a query selecting the keywords associated with resource *dataset_1*, and retrieving from a remote endpoint their human-readable representations in English, is the following:

```

1 PREFIX dcat: <http://www.w3.org/ns/dcat#>
2 SELECT ?keyw ?label
3 WHERE {
4   <http://.../dataset_1> dcat:keyword ?keyw .
5
6   SERVICE <http://some/endpoint> {
7     ?keyw skos:prefLabel ?label .
8     FILTER( LANG(?label) = "en" )
9   }
10 }
```

In particular, in the following of this paper, we are going to exploit query federation in order to reconstruct the normative XML representation of INSPIRE metadata from the corresponding RDF description.

2.2. The RITMARE Flagship Project

As already discussed in the introduction, adoption of our metadata management techniques does not require any prior knowledge on project RITMARE. However, this section outlines the data sharing infrastructure of this specific project as an example application of the methodology in a comprehensive architecture for the management of spatial data. RITMARE requires integration of all state-of-the-art contributions to Italian marine research into a coherent SDI [30], a framework for the collection and provisioning of geospatial data, metadata, networked services, and technologies. A coarse-grained categorisation of SDIs distinguishes between centralised and decentralised infrastructures, according to whether data and metadata are stored in a single repository or distributed among the distinct data providers. The RITMARE infrastructure belongs to the second kind, comprising:

- A set of peripheral nodes that expose standards-compliant metadata and services.
- A centralised catalogue service that provides access to the resources made available by the project as a whole.

The project is characterised by a heterogeneous set of data providers (public research bodies and inter-university consortia) as well as a variety of stakeholders (public administrations, private companies, and citizens). As a consequence, these entities envisage a varied corpus of heterogeneous data, metadata, workflows, and requirements. Beside this, data providers featured different degrees of maturity with regard to the provisioning of assets according to the mandated standards: This means that, in the development of the infrastructure, much effort was devoted to capacity building on the data provider-side. Moreover, being a national project, the RITMARE SDI is bound to the rules set by INSPIRE as well as by RNDT: Thus metadata management has a key role in the required architecture [31,32]. This has been achieved by providing a virtual appliance, the Geoinformation

Enabling ToolKIT software suite (<http://www.get-it.it/>), GET-IT for short, a FOSS product that is capable of kickstarting an autonomous node in the SDI for the collection, annotation, and deployment of geospatial data. Among the achievements of the GET-IT suite is integration of traditional geographic information (e.g., layers) with sensor data and, to our knowledge, GET-IT is the first application achieving this. Specifically, the GeoNode distribution (<http://geonode.org/>) has been complemented with ad-hoc components and, among these, the facilities for managing sensor descriptions and related observations in the Sensor Observation Service (SOS) (<http://www.opengeospatial.org/standards/sos>) implementation by 52°North (<http://52north.org/>). Geographic layers can then be mix-and-matched with real-time data from sensors [33,34].

2.3. Related Works

Among the many products available in the state of the art for the provision of geospatial metadata, the one provided by GeoNetwork version 2.8 and earlier (<http://geonetwork-opensource.org/>) is to our knowledge the only tool allowing for easily “pluggable” metadata schemas. Thus it is difficult to identify editors that can be compared to our solution: We can only consider the tools that, separately, address the metadata schemas that are more widely implemented in RITMARE, that is, ISO 191** profiles and SensorML descriptions. The use case presented in this work is focusing on the first category of metadata schemas, which is also the one most widely supported by editing tools; a good review of these is provided by the page maintained by the American FGDC (ISO metadata editor review: <https://www.fgdc.gov/metadata/iso-metadata-editor-review>). This source makes it apparent that, although more mature editors may provide features that are currently not supported by our tool, which is still in its infancy, existing ISO metadata editors only have partial support to customisation of the governing metadata schema and no support at all for third-party data sources. The same applies to the second category of metadata editing tools, that is, those devoted to SensorML. A similar review can be found on the OGC website (Sensor Web Enablement Software: http://www.ogcnetwork.net/SWE_Software). The two editing tools referred to in this review are the Pines SensorML Editor (<http://lxspine.googlepages.com/pine'ssensormleditor>), and the SensorML Process Editor (<http://code.google.com/p/sensorml-data-processing/>). Both are implementing an outdated version of SensorML and none implements the functionalities implied by our requirements. Also considering newer editing tools not included in this list, such as the OpenSensorHub SensorML editor (<https://github.com/opensensorhub/sensorml-editor>) and the SensorNanny - DrawMyObservatory editor (<https://github.com/ifremer/snanny-drawmyobservatory>), none of them provide such flexibility with respect to the metadata schema that is implemented. More importantly, plugging in data structures provided by third-parties is another functionality not implemented in existing editors.

2.4. EDI, a Template-Driven Metadata Editor

In developing GET-IT, we had to support creation of metadata compliant with RNDT and SensorML; we also wanted to support the core INSPIRE profile of ISO 19115/19119 as well as its transposition by different countries. Moreover, the heterogeneity of the user community in RITMARE demanded for a high degree of customisation, particularly with respect to project-specific context information. Hence, in order to manage such diversity, we decided to abstract from the specific output format and create a general-purpose tool that, appropriately parameterised by a custom metadata schema definition, could render a web-based authoring tool in the browser and assist the user in providing the metadata. EDI (<http://edidemo.get-it.it>) [35], the editing tool described in this paper, is constituted by a client-side JavaScript application that can autonomously create the interface and connect to the data sources that are specified in the given template. A server-side component, written in Java, executes the actual translation of the user input in both the XML and RDF representations that are supported by our architecture. To our knowledge, no existing tool provides the same functionalities with respect to the heterogeneity of metadata profiles that can be supported and the capability of

plugging in external data sources. The semantics-aware annotation of metadata fields is another characteristic that can not be found in state-of-the-art metadata editing tools.

The EDI metalanguage allows for tailoring the behaviour of the application to a specific project or domain. On the one hand, recourse to a metalanguage for expressing the target metadata schema allows for full tailoring of the latter to the application context in hand. On the other, integration of external data structures elicit customisation of the metadata property values that populate this schema. In RITMARE, such customisation starts with the creation of templates expressing the required metadata schemas (in our case, RNDT and SensorML). Secondly, context information associated with the project (e.g., the project's structure as collection of institutes and individuals as well as selected controlled vocabularies) have been formalised and made available as RDF data structures in order to be referred to in templates. Similarly, the applicable external data sources have been identified and plugged in the editing tool. Integration of EDI with the GET-IT suite also allowed for narrowing the number of required metadata fields by applying naming conventions on resource identifiers and by extracting information at data upload-time.

3. Metadata Management Scenario

Although we indulge in some technicalities in the Appendices, this paper is not meant to provide a comprehensive overview of the functionalities that are implemented by EDI (please refer to the documentation provided on GitHub for a developer view on the tool). However, customisation of the target metadata schema does not call for modification of a single line of code in the application. In fact, EDI is a template-driven metadata editor that, in principle, elicits any XML-based (or text-based) metadata schema: System administrators can either create a template from scratch or customise one of those provided with the application, which implement the basic INSPIRE profile, RNDT, and SensorML versions 1.0.1 and 2.0.0 profiled according to the OGC SOS Lightweight Profile [36]. In the following of this Section, we are going to focus on the features that, in line with the research problems addressed in the Introduction, allow for opening metadata management to the Web of Data.

Figure 1 presents the information flow regulated by our template structure, consisting of the following phases:

1. The system administrator executes the one-off creation or customisation of the template in order to set the metadata schema for the specific use case. In this phase, external data sources from the Web of Data can be plugged in: This allows metadata properties to refer to resources (persons, toponyms, code vales, etc.) that are managed by third parties. The template serves as input to EDI both in the peripheral and central nodes of RITMARE.
2. The metadata maintainer of a RITMARE peripheral node uses the editing interface that is created by EDI on the basis of template definitions and creates/edits metadata (the output of EDI). The data sources that have been plugged in the previous phase enable autocompletion functionalities that reduce as much as possible the effort required for metadata provision.
3. The metadata records in XML format are generated by EDI for insertion in catalogues and applications that understand the specific formalism, such as in the peripheral nodes of the RITMARE infrastructure. Typically, the entities referred to in the previous phase are rendered now as free-text property values.
4. The semantics-aware counterpart of the metadata description is also produced by EDI and stored as RDF data in the project's triple store (i.e., a database for RDF data). The record can also be published on the Web of Data and be accessed according to the same formats and protocols that allowed for plugging in external data sources in the first phase.
5. The end user can search the RITMARE central geoportal through the discovery client. When metadata records are requested according to the XML metadata schema, they are produced again by EDI on the basis of template definitions. Specifically, property values drawn from the data sources that are referred to in the template are accessed again at user request-time. This allows

for generating an XML description containing up-to-date property values, as detailed in the following of this Section.

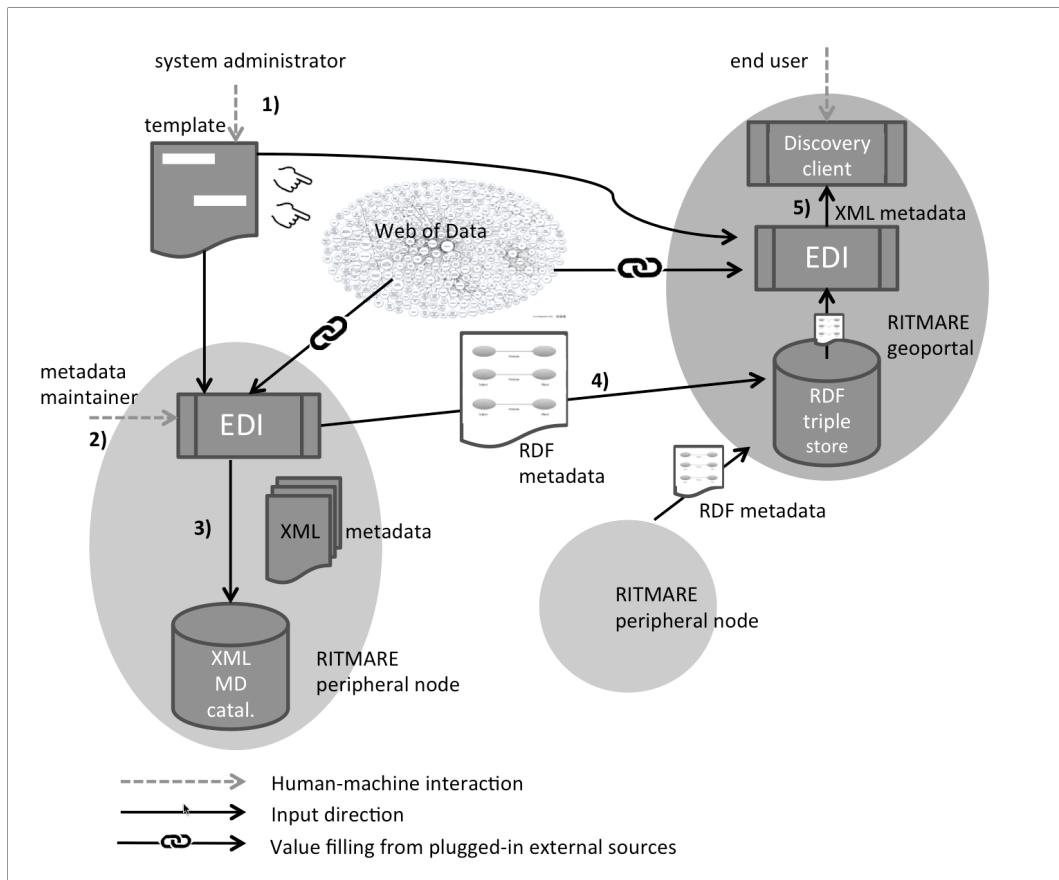


Figure 1. Metadata management in the RITMARE SDI.

3.1. Use Cases and Requirements

In this section, we introduce two use cases that may look straightforward to implement when tackled in a simplistic way but, as soon as new—perfectly reasonable—requirements pop in, may become far more challenging. In both use cases, the key point is selection of the data sources that are made accessible via the metadata editor. It should be noted that, although these use cases stem from the requirements posed by project RITMARE, they are significant in the broader scope of geospatial metadata management.

3.1.1. Specifying Points of Contact

Figure 2A shows the web form fragment that allows the metadata maintainer to specify a point of contact for a dataset according to the INSPIRE profile of ISO 19115/19119. As soon as the user starts typing the e-mail address, the interface suggests some options for completion of the metadata field (b). Upon selection of one of these, the following field describing the point of contact (that is, the field containing the name of the organisation the person works in) is automatically filled in (c). Also, the form provides a drop-down list for selecting the specific role the point of contact fulfils (in the example, “Resource provider”). One may argue that, with a supporting relational database behind the scenes, it is straightforward to implement exactly the same functionality. But what if the data structures that are leveraged on for autocompletion are managed by a third party? Although not apparent in this use case, this requirement (that is, integrating third-party data structures in the metadata editing application) is of utmost relevance for the following use case. Also, consider what

happens when some of the metadata fields involved in this use case (say, the e-mail address associated with the individual) change. Shall we rely upon inconsistent metadata featuring the old property values or try to keep them updated?

(A)

(B)

Figure 2. Use cases “specifying points of contact” (A); “keywords from controlled vocabularies” (B).

3.1.2. Keywords from Controlled Vocabularies

The web form fragment in Figure 2B allows the metadata maintainer to provide the keywords describing a dataset which, as often happens in geospatial metadata, are bound to a specific controlled vocabulary (or a set of). As an example, in INSPIRE-based metadata, the codelists provided by the reference ISO standards have been supplemented by those defined by the Directive (e.g., the codelist containing the INSPIRE Themes); also, distinct application domains may rely on further controlled vocabularies that are specific to a given CoP. In the use case, the codelist *Climate and Forecast Standard Names* has been selected in the comprehensive collection of thesauri defined by project SeaDataNet (<http://www.seadatanet.org/>) and provided through the NERC Vocabulary Server (<http://vocab.nerc.ac.uk/>) [37,38] and is used to find the suggestions that are proposed to the metadata maintainer (b). Depending on her choice, the accessory fields describing the keyword are automatically filled in (c).

Also here, a supporting relational database could provide the terminology grounding autocompletion of keywords; but what about the inevitable evolution of such terminology? It is inconvenient for system administrators to duplicate this data source and strive to keep it updated to the authoritative version. Rather, it would be better to directly link the application behaviour to

the aforementioned data source. What if, at some point, a “find similar assets” or “translate this metadata record” functionality is requested? In traditional discovery based on ISO 19136 metadata, implementing query expansion and multilingual functionalities constitutes a daunting task. Instead, it is straightforward to accomplish this with the URI-based metadata generated by our application.

In the next Section, we review the template components that allow for realising these use cases and also introduce the components that regulate translation of user input in either of the data formats we employ, respectively, for metadata representation and for XML encoding of the latter.

3.2. Template Structure

All activities related to metadata management, such as editing of fields and creation of the RDF and XML representations of metadata descriptions, are driven by the EDI template, in XML format as well, that is dependent on the specific schema that is required by a given project (e.g., RNDT, SensorML, etc.). The template specifies properties of metadata that shall translate into widgets in the editing interface: Some of these directly stem from regulations (e.g., whether a given field is mandatory or not, the associated multiplicity, etc.), while other properties depend on the context information associated with the specific project (such as the controlled vocabularies being used, the queries that allow the interface to propose default values for fields, etc.). Of course, the template also specifies the information required for translation into a valid XML document. It should be noted that the XML Schemas underlying the specific metadata format already contain some of this information. However, we found it unwieldy to leverage these specifications for articulating interface composition for a number of reasons. Firstly, many constraints are not directly stemming from the base schema but rather from the specific profile of the latter: As an example, both INSPIRE and RNDT rely on ISO 19136 for encoding, but each has its own mandatory fields, prescribed values, etc. Secondly, the most interesting aspects in this integration exercise, such as pervasive recourse to semantics-aware data structures and generation of multiple output formats, are not encoded in (and not the purpose of) the originating schemas.

As a consequence, a metalanguage for expressing these properties has been created and encoded as XML Schema. Figure 3 is showing the key components that are used in templates for defining a metadata field. Specifically, the *element* tag (note that we prefer term “tag” over “element” to avoid ambiguity with *element* components of templates) in Figure 3a defines individual metadata fields: The attributes defined provide a unique *xml:id* for it, specify whether it is mandatory or optional, and declare the associated multiplicity. Also, the field can be declared as *alternativeTo* another. The content of *element* tags include the multilingual visual cues that can be seen in the interface (the *label* and *help* tags) and the XPath location where multiple instances of the metadata field shall be rooted (the *hasRoot* tag). Tag *produces* contains the set of *items* that represent individual XML nodes that shall be created for the specific metadata field. Finally, tag *rdfOut* drives creation of the actual metadata representation to be stored in the RDF triple store; conversely, tag *rdfIn* drives extraction of metadata field values when re-creating the metadata description according to the schema the template is meant to implement. This tag contains a SPARQL query retrieving the necessary property values from the local triple store. Since multiple data sources may contribute to creating the final metadata, this query definition is only partial because it lacks the *triple patterns* (i.e., the query constraints) that shall be matched against the remote data sources that are defined. This aspect will be clarified in the following.

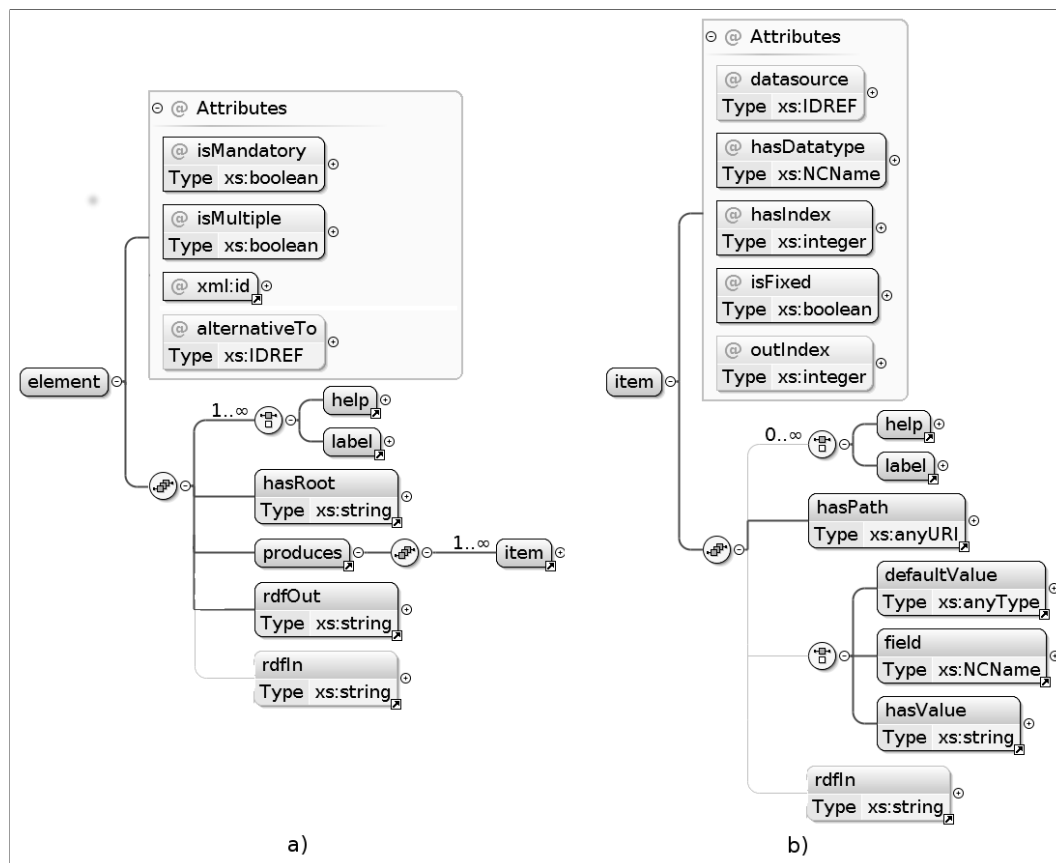


Figure 3. The *element* and *item* XML Schema components.

In fact, as anticipated above, *element* definitions can trigger the creation of multiple XML nodes (XML elements and attributes) in the target metadata file, which is the purpose of the *item* tag we are about to describe. This distinction is outmost necessary for the target format of our use cases because in ISO metadata (the reference schema the template structure was based upon) it is not infrequent to have the actual values selected by the metadata maintainer inserted in multiple places, together with fixed values in other positions in the specific XML sub-tree. Among others, this behaviour is apparent for keywords taken from controlled vocabularies where a single term choice by the metadata maintainer produces six different XML nodes in the final document (see Appendix D for an example of this). The semantics of the *item* tag is the following: Attributes *hasIndex* and *outIndex* specify the ordering of fields in the interface and in the output document, respectively. Attribute *isFixed* determines whether a widget shall be created in the editing interface (when set to “false”) or if the metadata field can be kept transparent to the end user because its value is known in advance. Other key attributes are *hasDatatype*, specifying the range of valid values for the item, and *datasource*, providing the aforementioned flexibility in the definition of external data sources. The tags included in individual *items* define, beside the visual cues we already found in the definition of the *element* tag, the specific value (when attribute *isFixed* is set to “true”), a default one, or the corresponding variable (the *field* tag) exposed by the associated *datasource*. The *hasPath* tag, specifying the XPath of the XML node that shall be created, is an essential component of *item* definitions. Finally, an optional *rdfIn* tag complements the SPARQL query in the *rdfIn* tag defined for the *element* as a whole. In fact, each *element* definition in the template defines the SPARQL query for retrieving the metadata property values that are required by the distinct *items*. Note that, since each of these may rely on a different endpoint, the SPARQL queries compiled from EDI templates are, by definition, federated. Appendix A contains the template definitions implementing the two use cases introduced in this Section.

Now we concentrate on the definition of data sources, the core feature for eliciting decentralised management of metadata. In fact, both use cases defined in Section 3.1 involve data sources: The first is constituted by the triple store we set up for managing context information in project RITMARE, while the second is provided by project SeaDataNet. Although managed by different parties, these data sources can be accessed in a seamless fashion, through an HTTP endpoint, because they share the same access protocol (SPARQL) and the same data format (RDF). Of course, the content of each of these may relate to schemata created with different formalisms (such as RDFS or OWL [39,40]) and with different purposes, but the commonality of the base format allows for univocal query mechanisms. In fact, both data sources are handled in the same way by EDI and either can be the target of *datasource* definitions in the template. Figure 4 shows the three different categories of data sources supported by templates; all of them share an *xml:id* attribute for unique identification and an *endpointType* attribute allowing administrators to plug in different triple stores by associating the correct request parameters. They also share the *url* tag allowing for per-datasource definition of endpoints (that is, the web addresses queries shall be posted to). The three categories are also described below:

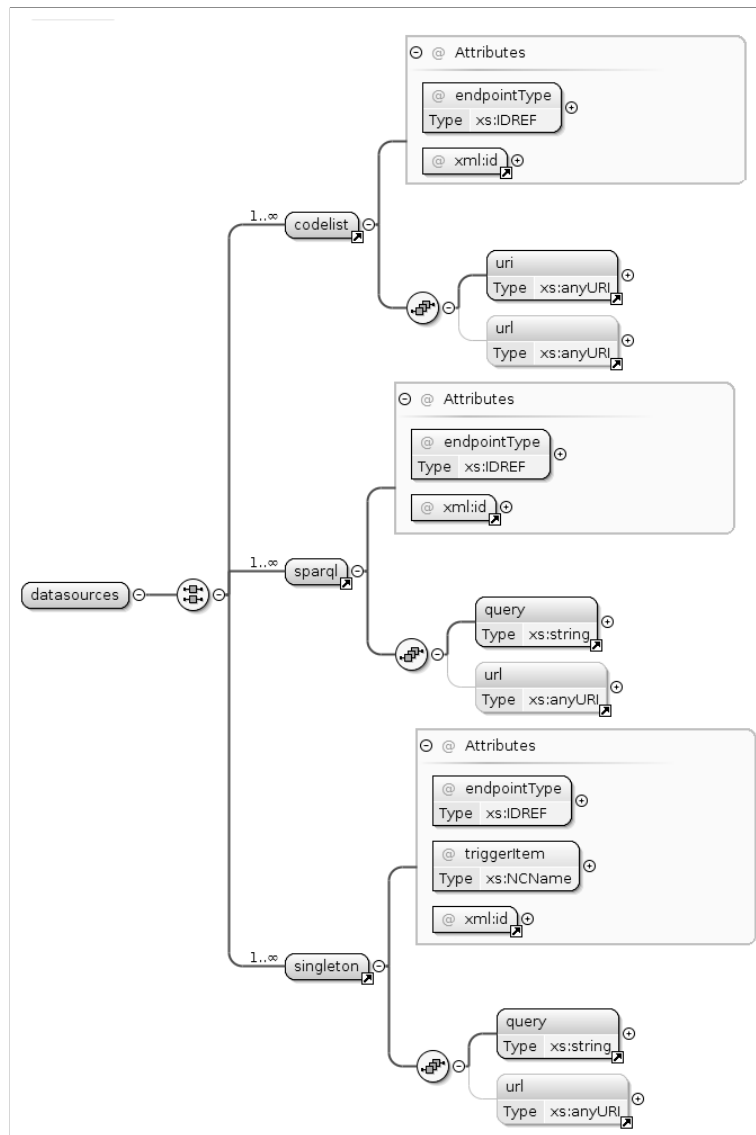


Figure 4. The *datasource* schema component.

Codelist: This category of datasources assumes that the nested *uri* tag refers to a SKOS thesaurus (a controlled vocabulary encoded according to this specific ontology) and executes a standard query for matching code values. This is the data source type allowing for creation of the drop-down list for selecting the role of a point of contact in Figure 2a.

Sparql: This category allows for executing generic SPARQL queries. This comes handy in the example use cases because the SeaDataNet endpoint provides SKOS-compliant thesauri whose structure slightly differs from that expected by the previous datasource type. Both use cases involve a datasource of this type.

Singleton: Executes queries that are required to return a single record (typically, as a consequence of a previous call to a datasource of the preceding type to provide autocompletion functionalities). Both use cases involve a datasource of this type.

Specifically, the use cases above rely on the RITMARE endpoint (<http://sparql.get-it.it/>) and the NERC Vocabulary Server endpoint (<http://vocab.nerc.ac.uk/sparql/>), for the data structures representing the user community and the controlled vocabularies, respectively. When accessed, these addresses provide a web form to issue queries to the endpoint; by providing the query as GET or POST request parameter, it is possible to issue queries in an automated way. As an example, the query that is used in use case “specifying points of contact” (matching the e-mail address of prospective points of contact) is shown in Listing 1.

Listing 1: SPARQL query for use case “specifying points of contact”

```

1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX vcard: <http://www.w3.org/2006/vcard/ns#>
3 PREFIX foaf: <http://xmlns.com/foaf/0.1/>
4 SELECT ?contact ?label
5 FROM <http://ritmare.it/rdfdata/lotrx>
6 WHERE {
7   ?contact rdf:type foaf:Person .
8   ?contact vcard:email ?label .
9   FILTER( REGEX( STR(?label), "$search_param", "i" ) )
10 }
11 }
12 ORDER BY ASC(?label)

```

The data structures that are looked up feature properties from the FOAF (<http://xmlns.com/foaf/spec/>) and vCard (<https://www.w3.org/TR/vcard-rdf/>) schemata: The first is used for modelling the researchers and institutes participating in RITMARE, the second provides fine-grained properties for detailing the former. In order to provide users of the EDI demo site with nonsensitive data to play with, we created example FOAF data structures for the characters in *The Lord of the Rings* and *The Hobbit*. By inserting the query in the web form provided by the first endpoint, it is possible to test the query before deploying the template: For doing this, the placeholder “\$search_param” shall be substituted with the actual search pattern, just like in Figure 2b. Then, the query is going to return the URIs and e-mail addresses of individuals matching the pattern.

A second query, shown in Listing 2, allows the application to autocomplete field “Institute” in the interface fragment shown in Figure 2a:

Listing 2: SPARQL query for use case “specifying points of contact”

```

1 PREFIX vcard: <http://www.w3.org/2006/vcard/ns#>
2 PREFIX foaf: <http://xmlns.com/foaf/0.1/>
3 SELECT ?inst
4 FROM <http://ritmare.it/rdfdata/lotrx>
5 WHERE {
6   <$search_param> vcard:org ?org .
7   ?org foaf:name ?inst .

```

```

8   FILTER( LANG(?inst) = "en")
9 }

```

By substituting any of the URIs returned by the first query to the placeholder “\$search_param”, the name of the corresponding organisation is returned.

Similarly, the second use case, “keywords from controlled vocabularies”, relies on the queries shown in the following Listings. Listing 3 is used to match user input against the *Climate and Forecast Standard Names* vocabulary P07 provided by SeaDataNet and this restriction can be seen in line 6; let aside this, the query is very similar to that in Listing 1.

Listing 3: SPARQL query for use case “keywords from controlled vocabularies”

```

1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
3 SELECT ?concept ?label
4 WHERE {
5   ?concept rdf:type skos:Concept .
6   <http://vocab.nerc.ac.uk/collection/P07/current/>
7     skos:member      ?concept .
8   ?concept skos:prefLabel ?label .
9   FILTER( LANG(?label) = "en" &&
10    REGEX( STR(?label), "$search_param", "i" ) )
11 }

```

The query in Listing 4 extracts the name of the thesaurus under consideration and the associated publication date. Since the format of the latter data value is a full timestamp, the BIND clause in line 8 extracts the fragment that matches the prescribed format yyyy-mm-dd and associates this value to variable *?date*:

Listing 4: SPARQL query for use case “keywords from controlled vocabularies”

```

1 PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
2 PREFIX dct: <http://purl.org/dc/terms/>
3 SELECT ?label ?date
4 WHERE {
5   ?voc skos:member      <$search_param> .
6   ?voc skos:prefLabel ?label .
7   ?voc dct:date        ?ts .
8   BIND( STRBEFORE(?ts, " ") AS ?date)
9 }

```

The template definitions for all data sources can be found in Appendix B and C.

3.3. Generating and Storing Metadata

Once metadata is posted to the server-side component of EDI, the prescribed XML output is generated by creating XML nodes following the XPath expressions defined by *hasRoot* and *hasPath* tags in the template. Additionally, the tag *rdfOut* defined by each *element* allows for specifying which RDF triples shall be produced and inserted in the triple store. In fact, during editing of metadata, EDI exposes (for fields that are not defined as fixed) the free-text values that are needed for generating the specific XML nodes but, under the hood, it keeps track of the URIs that have been selected by the user in order to create an RDF data fragment conveying the same semantics. As an example, the *rdfOut* tag in the first *element* defined in Appendix A (implementing the first use case, “specifying points of contact”) is depicted in Listing 5.

Listing 5: Content of the *rdfOut* tag associated with use case “specifying points of contact”

```

1 PREFIX dcat: <http://www.w3.org/ns/dcat#>
2 PREFIX vcard: <http://www.w3.org/2006/vcard/ns#>
3 INSERT
4 {
5   <$id_1_uri> dcat:contactPoint [
6     rdf:type      vcard:Individual ;
7     vcard:hasUID  <$resp_1_uri> ;
8     vcard:hasRole <$resp_3_uri>
9   ] .
10 }
11 WHERE {}

```

It relates the identifier (the URI) of the dataset being described (indicated as “\$id_1_uri”) to a *vcard:Individual* data structure via property *contactPoint* from the DCAT namespace: This data structure is composed of the identifier of the contact point and that of the specific role she covers (both expressed, again, as URIs). All queries in this Section contain placeholders following the production rule <element_id>_<item_index>[_uri]. They are substituted run-time with the actual values entered by the metadata maintainer. As an example, placeholder <\$resp_1_uri> is replaced with the URI associated with the first *item* in the *element* whose *xml:id* is “resp” (that is, the URI of the individual selected).

It should be noted that these triples are not in one-to-one correspondence with the XML nodes that are created in ISO metadata, a characteristics that is even more apparent in the *rdfOut* tag of the second *element* (implementing use case, “keywords from controlled vocabularies”), shown in Listing 6.

Listing 6: Content of the *rdfOut* tag associated with use case “keywords from controlled vocabularies”

```

1 PREFIX dcat: <http://www.w3.org/ns/dcat#>
2 INSERT
3 {
4   <$id_1_uri> dcat:keyword <$contr_voc_1_uri> .
5 }
6 WHERE {}

```

In this case, a single triple is sufficient to provide the semantics of the six *items* that are defined in the corresponding *element* in the template. At mere syntactic level, this is the major difference between our approach and any of the possible one-to-one translations of INSPIRE metadata into RDF, such as those in [41,42]. Section 4 is going to elaborate on the profound implications of our approach in metadata management. For the time being, it suffices to note that this practice is the enabling factor for the novel functionalities we are going to sketch in the next Section.

In fact, whenever the XML representation of a metadata record is requested, the template is parsed again in order to look up property values at metadata request-time. This is the role of *rdfIn* tags in the template: As an example, use case “specifying points of contact” defines three of such tags (see Appendix A) because retrieving the necessary data values may involve up to three distinct data sources, that is, the local endpoint we use for storing metadata, the endpoint hosting the information on contact points (and organisations), and finally the endpoint storing the codelists that define the available roles for the point of contact (only two endpoints in our worked-out example). Listing 7 shows the query that is compiled from this information:

Listing 7: SPARQL query originating from *rdfIn* tag definitions in use case “specifying points of contact”

```

1 PREFIX dcat: <http://www.w3.org/ns/dcat#>
2 PREFIX vcard: <http://www.w3.org/2006/vcard/ns#>
3 PREFIX foaf: <http://xmlns.com/foaf/0.1/>
4 PREFIX skos: <http://www.w3.org/2004/02/skos/core#>

```

```

5 SELECT ?1 ?2 ?3
6 WHERE {
7   <${id_1_uri}> dcat:contactPoint ?struct .
8   ?struct vcard:hasUID ?contact .
9   ?struct vcard:hasRole ?role .
10  SERVICE <http://sparql.get-it.it/> {
11    ?contact vcard:email ?1 .
12    ?contact vcard:org ?org .
13    ?org foaf:name ?2 .
14  }
15  SERVICE <http://sparql.get-it.it/> {
16    ?role skos:prefLabel ?3 .
17  }
18  FILTER( LANG(?2) = "en" && LANG(?3) = "en" )
19 }

```

Variable names in the projection clause are the numeric values identifying the distinct *items* of the specific *element*. The query is going to retrieve all *vcard:Individuals* related to the specific asset in locally stored metadata. The URIs identifying the person and her role as a contact point are then used in the federated queries indicated by *SERVICE* clauses. Each tuple in the result set will generate the corresponding XML data chunks as specified in the template.

Similarly, use case “keywords from controlled vocabularies” is going to entail federation of the NERC vocabulary server in order to retrieve the necessary keyword details:

Listing 8: Content of the *rdfln* tag associated with use case “keywords from controlled vocabularies”

```

1 PREFIX dcat: <http://www.w3.org/ns/dcat#>
2 PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
3 PREFIX dct: <http://purl.org/dc/terms/>
4 SELECT ?1 ?2 ?3
5 WHERE {
6   <${id_1_uri}> dcat:keyword ?keyw .
7   SERVICE <http://vocab.nerc.ac.uk/sparql/> {
8     ?keyw skos:prefLabel ?1 .
9     ?thes skos:member ?keyw .
10    ?thes skos:prefLabel ?2 .
11    ?thes dct:date ?timestamp .
12  }
13  BIND( STRBEFORE(?timestamp, " ") AS ?3)
14 }

```

The queries we compile by combining the content of *rdfln* tags are partly redundant with those provided for the definition of *datasources* but, for the time being, we did not find a viable means to express these queries in a unified way.

4. Discussion

In this Section, we check whether the metadata management practices that were proposed in the previous Sections are correctly addressing the issues and desiderata presented in Section 3.1. Moreover, we provide further ideas on how to extend the capabilities of state-of-the-art geoportals.

4.1. Assuring Metadata Consistency

The first use case put forward possible inconsistency issues that may arise as soon as one of the field values included in metadata changes. There is no acknowledged practice for (automatically) reflecting such changes in the metadata descriptions that annotate assets: Typically, human intervention is required to bring metadata descriptions back to consistency. Instead, the inherently deferred XML

production described in Section 3.3 is going to return up-to-date property values (of course, to the extent that the specific data source is kept updated). Note that this functionality can not be achieved with normative ISO-based INSPIRE metadata as well as with any one-to-one translation of these into RDF. To our knowledge, no existing metadata management technique can achieve automatic update of metadata fields, especially those that involve data provided by third parties.

4.2. Recommending Assets

The accessory information that can be drawn from heterogeneous data sources can be valuable for extending the scope of one's application. As an example, the URI of a point of contact in a metadata description may lead to people in her social network (indicated, for example, by *foaf:knows* property values) and these may point to relevant assets that did not match the discovery criteria (e.g., because of semantic heterogeneity issues). Moreover, when the user accessing the geoportal is authenticated, discovery results that have been deemed as relevant by people in her own social network (because the corresponding assets have been viewed, downloaded, bookmarked, etc.) can be suggested on the basis of search log analysis. Both these social networks (that of the point of contact and that of the user executing the search) could be further shortlisted by matching the preferred research topics of the individuals involved (expressed as *foaf:topic_interest* property values).

4.3. Expanding Queries

The advantages of relying on information drawn from the Web of Data are even more apparent when considering resources that are more semantically rich and connected with other resources of the same kind, such as terms in controlled vocabularies. Use case "keywords from controlled vocabularies" hinted at this when speculating on the "find similar assets" or "translate this metadata record" functionalities. The first can be implemented on the basis of the (poly)hierarchical structure of SKOS thesauri because the URI of a term can be used to derive more general, more specific, equivalent, or simply related terms. These can in turn increase the result set of a discovery by matching the URIs of these terms (or their text representations in traditional discovery) against catalogue metadata. When metadata is encoded as XML or text, the second functionality can only harness automatic translation tools; instead, the URIs of terms in controlled vocabularies are in principle language-neutral but the terms themselves can accommodate translations into multiple languages. Hence, a keyword from a controlled vocabulary can be straightforwardly translated into any of the supported languages.

4.4. Exploiting Gazetteer Information

Gazetteer services constitute the primary means to bridge the gap between geospatial and traditional (i.e., text-based) web searches. Even though the use cases presented in this paper refer to metadata that have no geographic connotation, our metadata descriptions cover the full set of INSPIRE metadata, comprising geographic and temporal extent (expressed as data structures complying with the schemata in [41]). Elaborating on the former of these properties, RDF-based gazetteer services, such as GeoNames (<http://www.geonames.org/>) can be used to "translate" text-based queries such as "salinity of water column 100 km West of Naples" into the corresponding geospatial query. Instead, geoportals typically require the user to provide coordinates or draw a rectangle (the "bounding box") on a map. Moreover, toponyms are typically organized as a hierarchy according to containment properties, thus allowing for more query expansion criteria.

4.5. Further Suggestions for Semantic Enrichment

The use cases in Section 3.1 entail data structures representing individuals, encoded according to the FOAF and vCard schemata, and descriptions of terms in controlled vocabularies, expressed as SKOS data structures. However, even a sneak peek at the Linked Open Data cloud diagram (<http://lod-cloud.net/>) can suggest a number of viable data sources for enriching metadata with semantic information. We acknowledge that relying on dependability of data sources that are out there

in the Web of Data, hence not under direct control of the system administrator customising EDI for a specific project, is a big leap of faith. However, in our opinion it is more advisable, on the long term, to refer to well-established data sources rather than to rely on proprietary data structures that are prone to be neglected as soon as a given project or initiative is discontinued. Please refer to [43] for a general view on the RDF data structures we integrated in project RITMARE for the purpose of grounding metadata management.

5. Conclusions and Future Work

In this paper, we presented a scenario illustrating the methodology for metadata management that has been set up for project RITMARE. It hinges on a client-server application rendering the editing interface for a given metadata schema and taking care of the generation of XML and RDF descriptions on the basis of the data inserted by the metadata maintainer. This tool, EDI, is a FOSS package (the RITMARE SP7 team is part of the GEOforALL network: <http://www.geoforall.org/>) that can be easily customised to the requirements of other projects. If compliance with one of the supported profiles is sufficient to a given application context, the software can be used out-of-the-box. A second step in customisation of the tool is constituted by personalisation of the template that regulates its behaviour: This is assisted by the XML Schema associated with the templating language, whose essentials have been presented in this work. In the future, further templates may be added to the GitHub distribution (<https://github.com/SP7-Ritmare>) as the output of a community effort. Finally, external data sources can be easily plugged in for full customisation of the metadata editing experience: We exemplified either recourse to proprietary data sources describing a project (e.g., describing users, institutes, roles, etc.) and the integration of third-party data structures provided as SPARQL endpoints (e.g., to leverage controlled vocabularies in use by the specific CoP).

This feature is aimed at fostering a decentralised paradigm for metadata management: In our framework, attribute values are retrieved from the (possibly remote) data structures that are associated with the specific principal. As an example, the e-mail address of a point of contact is directly taken from her FOAF description, thus avoiding redundancies and inconsistencies. Our methodology has the key advantage of enabling automatic update of metadata attributes that have URIs (directly or indirectly) associated with them: In our opinion, this is sufficient to deem our metadata management paradigm as more robust with respect to traditional centralised strategies. In fact, centralised strategies cannot but repeat the same property values (again, consider the e-mail address of a point of contact) over and over in metadata records, hampering efficient propagation of updates and impeding more fine-grained management of metadata items.

There is also a number of novel query expansion techniques that can be based on the RDF data structures that express metadata: These can identify people, keywords, toponyms, and all categories of descriptions one can find on the Web of Data. The extent to which this can impact on precision and recall in discovery of geospatial assets (an evaluation out of the scope of this work) is yet to be determined and compared with other strategies for semantics-aware discovery, such as those in [44,45]. However, an apparent advantage of our methodology with respect to the first approach is that no preemptive processing of metadata descriptions is required, because semantic lift is implicit in metadata editing. Instead, in [44] traditional ISO-based metadata is first processed in order to (heuristically) associate property values with URIs. Semantic lift still makes sense for query parameters entered by the end user at discovery-time, as in [45], and this is one of the aspects we are going to implement in the interface of the RITMARE geoportal.

As a bottom line, we point out that the decisions that drove development of the methodology described in this paper find their roots in our vision of geospatial metadata as multi-tenanted descriptions. Unfortunately, metadata are typically overlooked, not exhaustive, or inconsistent. Then it makes sense to encompass third-party data structures if this means prolonging the time lapse in which metadata descriptions can be relied upon. Sourcing these external data structures as Open Data does not pose any condition on the usage or open-ness of the assets referred to in metadata descriptions.

Still, sharing data and metadata in an open fashion is a recommended practice we generally subscribe to: In particular, in the context of RITMARE, we promoted an open data policy [46] that can easily match those characterising Open Science initiatives [47–49].

Future activities in this research thread are going to focus on generalising the approach. Specifically, we are going to generate RDF-based descriptions from ISO metadata that have been produced by tools other than EDI, i.e., devoid of any semantic reference. In the beginning, the template language introduced in this paper is going to retain its importance because semantic lift requires definition of data sources and SPARQL queries, such as those in the implementation of the use cases presented in this work. Then, loosening this requirement is going to be the final step toward a generally applicable framework for decentralised, semantics-aware metadata management.

Acknowledgments: The activities described in this paper have been funded by projects RITMARE and LifeWatch.

Author Contributions: Fugazza and Oggioni conceived and designed the application. Pavesi developed EDI with support by Pepe, for the INSPIRE and RNDT templates, Oggioni and Tagliolato, for the SensorML templates. Carrara coordinated the research group. Fugazza wrote the paper and reviewed it with Pepe, Tagliolato, and Carrara. All authors read and approved the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CoP	Community of Practice
FGDC	Federal Geographic Data Committee
FOAF	Friend Of A Friend
FOSS	Free and Open Source Software
INSPIRE	INfrastructure for SPatial InfoRmation in Europe
RDF	Resource Description Framework
RITMARE	Ricerca Italiana per il MARE - Italian research for the sea
RNDT	Repertorio Nazionale dei Dati Territoriali—national repository of territorial data
SDI	Spatial Data Infrastructure
SPARQL	SPARQL Protocol and RDF Query Language
SWE	Sensor Web Enablement
Turtle	Terse RDF Triple Language
URI	Uniform Resource Identifiers
XML	eXtensible Markup Language

Appendix A. Template Definitions for the Use Cases in Section 3

```

1 <element xml:id="resp" isMandatory="true" isMultiple="true">
2   <label xml:lang="en">Responsible party</label>
3   <help xml:lang="en">...</help>
4   <hasRoot>/gmd:MD_Metadata/.../gmd:MD_DataIdentification</hasRoot>
5   <produces>
6     <item hasIndex="1" outIndex="2" isFixed="false"
7       hasDatatype="autoCompletion" datasource="person">
8       <label xml:lang="en">Email</label>
9       <hasPath>/.../gmd:electronicMailAddress/...</hasPath>
10      <RDFin><![CDATA [
11        ?contact vcard:email ?1 .
12        ?contact vcard:org ?org .
13        ?org foaf:name ?2 .
14      ]]></RDFin>
15    </item>
16    <item hasIndex="2" outIndex="1" isFixed="false"

```



```

17     hasDatatype="select" datasource="personS">
18     <label xml:lang="en">Institute</label>
19     <hasPath>/.../gmd:organisationName/...</hasPath>
20 </item>
21 <item hasIndex="4" isFixed="false"
22     hasDatatype="codelist" datasource="roleCodes">
23     <label xml:lang="en">Role</label>
24     <hasPath>/.../gmd:CI_RoleCode/@codeListValue</hasPath>
25     <defaultValue>http://.../resourceProvider</defaultValue>
26     <RDFin><![CDATA[
27         ?role      skos:prefLabel  ?3 .
28     ]]></RDFin>
29 </item>
30 </produces>
31 <RDFout><![CDATA[
32     PREFIX dcat: <http://www.w3.org/ns/dcat#>
33     PREFIX vcard: <http://www.w3.org/2006/vcard/ns#>
34     INSERT
35     {
36         <$id_1_uri> dcat:contactPoint [
37             rdf:type      vcard:Individual ;
38             vcard:hasUID  <$resp_1_uri> ;
39             vcard:hasRole <$resp_3_uri>
40         ] .
41     }
42     WHERE {}
43 ]]></RDFout>
44 <RDFin><![CDATA[
45     PREFIX dcat: <http://www.w3.org/ns/dcat#>
46     PREFIX vcard: <http://www.w3.org/2006/vcard/ns#>
47     PREFIX foaf: <http://xmlns.com/foaf/0.1/>
48     PREFIX skos: <http://.../skos/core#>
49     SELECT ?1 ?2 ?3
50     WHERE {
51         <$id_1_uri> dcat:contactPoint ?struct .
52         ?struct    vcard:hasUID      ?contact .
53         ?struct    vcard:hasRole     ?role .
54         FILTER( LANG(?2) = "en" && LANG(?3) = "en" )
55     }
56 ]]></RDFin>
57 </element>
58
59 <element xml:id="contr_voc" isMandatory="false" isMultiple="true">
60 <label xml:lang="en">Keyword from controlled vocabularies</label>
61 <help xml:lang="en">...</help>
62 <hasRoot>/gmd:MD_Metadata/.../gmd:MD_DataIdentification</hasRoot>
63 <produces>
64     <item hasIndex="1" isFixed="false"
65         hasDatatype="autoCompletion" datasource="keyw_SDN">
66         <label xml:lang="en">Keyword</label>
67         <hasPath>/.../gmd:keyword/...</hasPath>
68         <RDFin><![CDATA[
69             ?keyw  skos:prefLabel  ?1 .
70             ?thes  skos:member     ?keyw .
71             ?thes  skos:prefLabel  ?2 .

```

```

72         ?thes dct:date          ?timestamp .
73         BIND( STRBEFORE(?timestamp, " ") AS ?3)
74     ]]></RDFin>
75 </item>
76 <item hasIndex="2" isFixed="false"
77     hasDatatype="select" datasource="keyw_SDN_S">
78     <label xml:lang="en">Originating controlled vocabulary</label>
79     <hasPath>/.../gmd:title/...</hasPath>
80     <field>label</field>
81 </item>
82 <item hasIndex="3" isFixed="false"
83     hasDatatype="select" datasource="keyw_SDN_S">
84     <label xml:lang="en">Publication date (yyyy-mm-dd)</label>
85     <hasPath>/.../gco:Date</hasPath>
86     <field>date</field>
87 </item>
88     ...
89 </produces>
90 <RDFout><![CDATA[
91     PREFIX dcat: <http://www.w3.org/ns/dcat#>
92     INSERT
93     {
94         <$id_1_uri> dcat:keyword <$contr_voc_1_uri> .
95     }
96     WHERE {}
97 ]]></RDFout>
98 <RDFin><![CDATA[
99     PREFIX dcat: <http://www.w3.org/ns/dcat#>
100    PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
101    PREFIX dct: <http://purl.org/dc/terms/>
102    SELECT ?1 ?2 ?3
103    WHERE {
104        <$id_1_uri> dcat:keyword ?keyw .
105    }
106 ]]></RDFin>
107 </element>

```

Appendix B. Data Sources for Use Case “Specifying Points of Contact”

```

1 <sparql xml:id="person" endpointType="virtuoso">
2 <query><![CDATA[
3     PREFIX vcard: <http://www.w3.org/2006/vcard/ns#>
4     SELECT ?contact ?label
5     FROM <http://ritmare.it/rdfdata/lotrx>
6     WHERE {
7         ?contact rdf:type foaf:Person .
8         ?contact vcard:email ?label .
9         FILTER( REGEX( STR(?label), "$search_param", "i") )
10    }
11    ORDER BY ASC(?label)
12 ]]></query>
13 </sparql>
14
15 <singleton xml:id="personS" endpointType="virtuoso"
16     triggerItem="resp_1_uri">

```

```

17 <query><![CDATA [
18   PREFIX vcard: <http://www.w3.org/2006/vcard/ns#>
19   SELECT ?inst
20   FROM <http://ritmare.it/rdfdata/lotrx>
21   WHERE {
22     <$search_param> vcard:org ?org.
23     ?org foaf:name ?inst .
24     FILTER(LANG(?inst)='en')
25   }
26 ]]></query>
27 </singleton>
28
29 <odelist xml:id="roleCodes" endpointType="virtuoso">
30   <uri>http://.../ResponsiblePartyRole</uri>
31 </odelist>

```

Appendix C. Data Sources for Use Case “Keywords from Controlled Vocabularies”

```

1 <sparql xml:id="keyw_SDN" endpointType="fuseki">
2 <query><![CDATA [
3   PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
4   PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
5   SELECT ?concept ?label
6   WHERE {
7     ?concept rdf:type skos:Concept.
8     <http://vocab.nerc.ac.uk/collection/P07/current/>
9       skos:member ?concept .
10    ?concept skos:prefLabel ?label .
11    FILTER(
12      LANG(?label)="en" && REGEX(STR(?label),"$search_param","i")
13    )
14  }
15 ]]></query>
16 <url>http://vocab.nerc.ac.uk/sparql/sparql</url>
17 </sparql>
18
19 <singleton xml:id="keyw_SDN_S" endpointType="fuseki"
20   triggerItem="contr_voc_1_uri">
21 <query><![CDATA [
22   PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
23   PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
24   PREFIX dct: <http://purl.org/dc/terms/>
25   SELECT ?label ?date
26   WHERE {
27     ?voc skos:member <$search_param> .
28     ?voc skos:prefLabel ?label .
29     ?voc dct:date ?ts .
30     BIND( STRBEFORE(?ts, " ") AS ?date)
31   }
32 ]]></query>
33 <url>http://vocab.
34   .ac.uk/sparql/sparql</url>
35 </singleton>

```

Appendix D. ISO Metadata Fragments Produced by the Two Use Cases

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <gmd:MD_Metadata
3      xmlns:gmd="http://www.isotc211.org/2005/gmd"
4      xmlns:gco="http://www.isotc211.org/2005/gco"
5      xmlns:xlink="http://www.w3.org/1999/xlink"
6      xmlns:gml="http://www.opengis.net/gml/3.2"
7      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
8      xsi:schemaLocation="http://www.isotc211.org/2005/gmd
9      http://schemas.opengis.net/iso/19139/20060504/gmd/gmd.xsd">
10     ...
11     <gmd:contact>
12         <gmd:CI_ResponsibleParty>
13             <gmd:organisationName>
14                 <gco:CharacterString>
15                     Fellowship of the Ring
16                 </gco:CharacterString>
17             </gmd:organisationName>
18             <gmd:contactInfo>
19                 <gmd:CI_Contact>
20                     <gmd:address>
21                         <gmd:CI_Address>
22                             <gmd:electronicMailAddress>
23                                 <gco:CharacterString>
24                                     mailto:boromir.son-denethor@lotr.org
25                                 </gco:CharacterString>
26                             </gmd:electronicMailAddress>
27                         </gmd:CI_Address>
28                     </gmd:address>
29                 </gmd:CI_Contact>
30             </gmd:contactInfo>
31             <gmd:role>
32                 <gmd:CI_RoleCode
33                     codeList="http://.../gmxCodelists.xml#CI_RoleCode"
34                     codeListValue="pointOfContact">
35                     pointOfContact
36                 </gmd:CI_RoleCode>
37             </gmd:role>
38         </gmd:CI_ResponsibleParty>
39     </gmd:contact>
40     <gmd:descriptiveKeywords>
41         <gmd:MD_Keywords>
42             <gmd:keyword>
43                 <gco:CharacterString>
44                     19'-butanoyloxyfucoxanthin
45                 </gco:CharacterString>
46             </gmd:keyword>
47             <gmd:thesaurusName>
48                 <gmd:CI_Citation>
49                     <gmd:title>
50                         <gco:CharacterString>
51                             Marisaurus Thesaurus
52                         </gco:CharacterString>
53                     </gmd:title>

```

```

54         <gmd:date>
55             <gmd:CI_Date>
56                 <gmd:date>
57                     <gco:Date>2010-09-22</gco:Date>
58                 </gmd:date>
59                 <gmd:dateType>
60                     <gmd:CI_DateTypeCode>
61                         codeList="http://...#CI_DateTypeCode"
62                         codeListValue="publication">
63                             publication
64                     </gmd:CI_DateTypeCode>
65                 </gmd:dateType>
66             </gmd:CI_Date>
67         </gmd:date>
68     </gmd:CI_Citation>
69 </gmd:thesaurusName>
70 </gmd:MD_Keywords>
71 </gmd:descriptiveKeywords>
72 ...
73 </gmd:MD_Metadata>

```

References

1. De Smith, M.J.; Goodchild, M.F.; Longley, P. *Geospatial Analysis: A Comprehensive Guide to Principles, Techniques and Software Tools*; Troubador Publishing Ltd.: Leicester, UK, 2007.
2. Khare, R.; Rifkin, A. XML: Modeling Data and Metadata. In Proceedings of the ACM CSCW98 Conference on Computer Supported Cooperative Work, Seattle, WA, USA, 14–18 November 1998; p. 430.
3. RDM Working Group. *Reference Data and Metadata Position Paper*; 2002; p. 45. Available online: http://inspire.jrc.ec.europa.eu/reports/position_papers/inspire_rdm_pp_v4_3_en.pdf (accessed on 30 November 2016).
4. European Commission. *Directive 2007/2/EC of the European Parliament and of the Council of 14 March 2007 Establishing an Infrastructure for Spatial Information in The European Community (INSPIRE)*; Technical Report; 2007. Available online: <http://eur-lex.europa.eu/legal-content/EN/ALL/?uri=CELEX:32007L0002> (accessed on 30 November 2016).
5. European Commission. *Commission Regulation (EC) No 1205/2008 of 3 December 2008 Implementing Directive 2007/2/EC of the European Parliament and of the Council as Regards Metadata*; Technical Report; 2008. Available online: <http://eur-lex.europa.eu/legal-content/EN/ALL/?uri=CELEX:32008R1205> (accessed on 30 November 2016).
6. European Commission. *Corrigendum to Commission Regulation (EC) No 1205/2008 of 3 December 2008 Implementing Directive 2007/2/EC of the European Parliament and of the Council as Regards Metadata (OJ L 326, 4.12.2008)*; Technical Report; 2008. Available online: <http://eur-lex.europa.eu/legal-content/EN/ALL/?uri=CELEX:32008R1205R%2802%29> (accessed on 30 November 2016).
7. ISO. *ISO 19115:2014 Geographic Information—Metadata*; Standard, International Organization for Standardization (TC 211), 2014. Available online: http://www.iso.org/iso/catalogue/catalogue_ics/catalogue_detail_ics.htm?csnumber=53798 (accessed on 30 November 2016).
8. ISO. *ISO 19119:2005 Geographic Information—Services*; Standard, International Organization for Standardization (TC 211), 2005. Available online: http://www.iso.org/iso/catalogue_detail.htm?csnumber=39890 (accessed on 30 November 2016).
9. ISO. *ISO 19136:2007 Geographic Information—Geography Markup Language (GML)*; Standard, International Organization for Standardization (TC 211), 2007. Available online: http://www.iso.org/iso/catalogue/catalogue_tc/catalogue_detail.htm?csnumber=32554 (accessed on 30 November 2016).
10. INSPIRE Thematic Working Group on Environmental Monitoring. *D2.8.III.7 INSPIRE Data Specification on Environmental Monitoring Facilities—Draft Guidelines*; European Commission Joint Research Centre: Petten, The Netherlands, 2011; p. 22.

11. INSPIRE Thematic Working Group on Species Distribution. *D2.8.III.19 INSPIRE Data Specification on Species Distribution – Technical Guidelines*; European Commission Joint Research Centre: Petten, The Netherlands, 2013; p. 22.
12. Berners-Lee, T.; Hendler, J.; Lassila, O. *The Semantic Web*; Scientific American: New York, NY, USA, 2001; Volume 284, pp. 34–43.
13. Bizer, C.; Heath, T.; Berners-Lee, T. Linked data—the story so far. In *Semantic Services, Interoperability and Web Applications: Emerging Concepts*; IGI Global: Hershey, PA, USA 2009; pp. 205–227.
14. Garcia, R.; Celma, O. Semantic integration and retrieval of multimedia metadata. In Proceedings of the 5th International Workshop on Knowledge Markup and Semantic Annotation, Bournemouth, UK, 9–11 October 2006; pp. 69–80.
15. Kurki, T.; Jokela, S.; Sulonen, R.; Turpeinen, M. Agents in delivering personalized content based on semantic metadata. In Proceedings of the 1999 AAAI Spring Symposium Workshop on Intelligent Agents in Cyberspace, Stanford, CA, USA, 22–24 March 1999; pp. 84–93.
16. Heß, A.; Kushmerick, N. Learning to attach semantic metadata to web services. In *The Semantic Web-ISWC 2003*; Springer: Berlin, Germany, 2003; pp. 258–273.
17. AgID. *Repertorio Nazionale dei Dati Territoriali—RNDT*; Technical Report, Agenzia per l'Italia Digitale (AgID); 2012. Available online: <http://archivio.digitpa.gov.it/repertorio-nazionale-dei-dati-territoriali-rndt> (accessed on 30 November 2016).
18. Open Geospatial Consortium. OpenGIS Sensor Model Language (SensorML) Implementation Specification. In *Design*; Open Geospatial Consortium: Wayland, MA, USA, 2007; p. 180.
19. OGC. *OGC® SensorML: Model and XML Encoding Standard*; Encoding Standard OGC-12-000; Open Geospatial Consortium: Wayland, MA, USA, 2014.
20. Bishr, Y.A. Overcoming the semantic and other barriers to GIS interoperability. *Int. J. Geogr. Inf. Sci.* **1998**, *12*, 229–314.
21. Lassila, O.; Hendler, J. *Embracing Web 3.0*; IEEE Internet Computing: Baltimore, MD, USA, 2007; Volume 11, pp. 90–93.
22. Reed, C.; Botts, M.; Davidson, J. *OGC® Sensor Web Enablement: Overview and High Level Architecture*; 2007 IEEE Autotestcon: Baltimore, MD, USA, 2007; pp. 372–380. doi:10.1109/AUTEST.2007.4374243
23. Cyganiak, R.; Wood, D.; Lanthaler, M. *RDF 1.1 Concepts and Abstract Syntax—W3C Recommendation 25 February 2014*; Technical Report; 2014. Available online: <http://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/> (accessed on 30 November 2016).
24. Hitzler, P.; Krtzsch, M.; Rudolph, S. *Foundations of Semantic Web Technologies*, 1st ed.; Chapman & Hall/CRC: Boca Raton, FL, USA, 2009.
25. Berners-Lee, T.; Fielding, R.; Masinter, L. *Uniform Resource Identifier (URI): Generic Syntax*; Technical Report; 2005. Available online: <https://tools.ietf.org/html/rfc3986> (accessed on 30 November 2016).
26. World Wide Web Consortium. *Namespaces in XML 1.0 (Third Edition)*; Technical Report; 2009. Available online: <https://www.w3.org/TR/REC-xml-names/> (accessed on 30 November 2016).
27. Beckett, D.; Berners-Lee, T.; Prud'hommeaux, E.; Carothers, G. *RDF 1.1 Turtle—Terse RDF Triple Language*; Technical Report; 2014. Available online: <https://www.w3.org/TR/turtle/> (accessed on 30 November 2016).
28. Harris, S.; Seaborne, A. *SPARQL 1.1 Query Language—W3C Recommendation 21 March 2013*; Technical Report; 2013. Available online: <http://www.w3.org/TR/sparql11-query/> (accessed on 30 November 2016).
29. Cambridge Semantics. *SPARQL vs. SQL*; Technical Report. Available online: <http://www.cambridgesemantics.com/semantic-university/sparql-vs-sql-intro> (accessed on 30 November 2016).
30. Hjelmager, J.; Moellering, H.; Cooper, A.K.; Delgado, T.; Rajabifard, A.; Rapant, P.; Danko, D.M.; Huet, M.; Laurent, D.; Aalders, H.; et al. An initial formal model for spatial data infrastructures. *Int. J. Geogr. Inf. Sci.* **2008**, *22*, 1295–1309.
31. Tagliolato, P.; Oggioni, A.; Fugazza, C.; Pepe, M.; Carrara, P. Sensor metadata blueprints and computer-aided editing for disciplined SensorML. *IOP Conf. Ser. Earth Environ. Sci.* **2016**, *34*, 012036.
32. Fugazza, C.; Pepe, M.; Oggioni, A.; Tagliolato, P.; Carrara, P. Streamlining geospatial metadata in the Semantic Web. *IOP Conf. Ser. Earth Environ. Sci.* **2016**, *34*, 012009.
33. Fugazza, C.; Menegon, S.; Pepe, M.; Oggioni, A.; Carrara, P. The RITMARE Starter Kit: Bottom-up capacity building for geospatial data providers. In Proceedings of the 9th International Conference on Software Paradigm Trends (ICSOFT-PT), Vienna, Austria, 29–31 August 2014; pp. 169–176.

34. Oggioni, A.; Tagliolato, P.; Fugazza, C.; Pepe, M.; Menegon, S.; Pavesi, F.; Carrara, P. *Oceanographic and Marine Cross-Domain Data Management for Sustainable Development*; Diviacco, P., Leadbetter, A., Glaves, H., Eds.; IGI Global: Hershey, PA, USA, 2017; pp. 200–223.
35. Pavesi, F.; Basoni, A.; Fugazza, C.; Menegon, S.; Oggioni, A.; Pepe, M.; Tagliolato, P.; Carrara, P. EDI—A template-driven metadata editor for research data. *J. Open Res. Softw.* **2016**, doi:10.5334/jors.106.
36. Open Geospatial Consortium. *OGC Best Practice for Sensor Web Enablement Lightweight SOS Profile for Stationary In-Situ Sensors—OGC 11-169r1*; Technical Report; Open Geospatial Consortium: Wayland, MA, USA, 2014.
37. Leadbetter, A.; Lowry, R.; Clements, D. *The NERC Vocabulary Server: Version 2.0*; Geophysical Research Abstracts; EGU—European Geosciences Union GmbH: Munich, Germany, 2012; Volume 14.
38. Leadbetter, A. Linked Ocean Data. In *The Semantic Web in Earth and Space Science. Current Status and Future Directions*; AKA GmbH: Berlin, Germany, 2015; pp. 11–31.
39. RDF Working Group. *RDF Schema 1.1*; Technical Report; 2014. Available online: <https://www.w3.org/TR/rdf-schema/> (accessed on 30 November 2016).
40. OWL Working Group. *Web Ontology Language (OWL)*; Technical Report; 2012. Available online: <https://www.w3.org/2001/sw/wiki/OWL> (accessed on 30 November 2016).
41. Simon Cox. OWL Representation of the ISO/TC 211 Harmonized UML Model for Geographic Information. Technical Report, 2013. Available online: <http://def.seegrid.csiro.au/static/isotc211/iso19115/2003/> (accessed on 30 November 2016).
42. European Commission. *GeoDCAT-AP v1.0*; Technical Report; 2015. Available online: https://joinup.ec.europa.eu/asset/dcat_application_profile/asset_release/geodcat-ap-v10 (accessed on 30 November 2016).
43. Fugazza, C.; Pepe, M.; Oggioni, A.; Pavesi, F.; Carrara, P. A holistic, semantics-aware approach to Spatial Data Infrastructures. In *Proceedings of the 3rd International Conference on Data Management Technologies and Applications (DATA)*, Vienna, Austria, 29–31 August 2014; pp. 349–356.
44. Fugazza, C.; Luraschi, G. Semantics-Aware Indexing of Geospatial Resources Based on Multilingual Thesauri: Methodology and Preliminary Results. *Int. J. Spat. Data Infrastruct. Res.* **2012**, *7*, 16–37.
45. Santoro, M.; Mazzetti, P.; Nativi, S.; Fugazza, C.; Granell, C.; Díaz, L. Methodologies for augmented discovery of geospatial resources. In *Discovery of Geospatial Resources: Methodologies, Technologies, and Emergent Applications*; Díaz, L., Granell, C., Huerta, J., Eds.; IGI Global: Hershey, PA, USA, 2012; pp. 172–203.
46. Documento Per La Definizione Di Una Politica Nella Gestione E Utilizzo Dei Dati E Dei Prodotti Resi Disponibili Nell'ambito Del Progetto RITMARE. Available online: <http://www.ritmare.it/area-download?download=187:data-policy-new> (accessed on 30 November 2016).
47. Reichman, O.; Jones, M.B.; Schildhauer, M.P. Challenges and opportunities of open data in ecology. *Science* **2011**, *331*, 703–705.
48. David, P.A. The economic logic of open science and the balance between private property rights and the public domain in scientific data and information: A primer. In *The role of scientific and technical data and information in the public domain: Proceedings of a Symposium*; The National Academies Press: Washington, DC, USA, 2003; pp. 19–34.
49. Uhler, P.F.; Schröder, P. Open data for global science. *Data Sci. J. CODATA* **2007**, *6*, 36–53.



© 2016 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) license (<http://creativecommons.org/licenses/by/4.0/>).