Project No. 249024

NETMAR

Open service network for marine environmental data

Instrument: *Please tick*	CA	✓ **STREP**	IP	NOE

ICT - Information and Communication Technologies Theme

D4.1 – Review of Semantic Frameworks

Reference: D4.1_Review_Semantic_Frameworks_r1_20100609

Due date of deliverable (as in Annex 1): M0 + 3
Actual submission date: 9 June 2010

Start date of project: 1 March 2010 Duration: 3 years

Coastal and Marine Resources Centre (CMRC), University College Cork, National University of Ireland

Revision 1

Project co-funded by the European Commission within the Seventh Framework Programme (2007-2013)		
Dissemination Level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

European Commission
Information Society and Media

NETMAR

Open service network for marine environmental data
Project Reference: 249024
Contract Type: Collaborative Project
Start/End Date: 01/03/2010 - 31/01/2013
Duration: 36 months

Coordinator: Prof. Stein Sandven

Nansen Environmental and Remote Sensing Center
Thormøhlensgate 47, Bergen, Norway
Tel.: +47-55205800
Fax. +47 55 20 58 01
E-mail: stein.sandven@nersc.no

Acknowledgements

The work described in this report has been partially funded by the European Commission under the Seventh Framework Programme, Theme ICT 2009.6.4 ICT for environmental services and climate change adaptation.

Consortium

The NETMAR Consortium is comprised of:

- Nansen Environmental and Remote Sensing Center (NERSC), Norway (coordinator).
  Project Coordinator: Prof. Stein Sandven (stein.sandven@nersc.no)
  Deputy Coordinator: Dr. Torill Hamre (torill.hamre@nersc.no)
  Quality Control Manager. Mr Lasse H. Pettersson (lasse.pettersson@nersc.no)
- British Oceanographic Data Centre (BODC), National Environment Research Council, United Kingdom
  Contact: Dr Roy Lowry (rkl@bodc.ac.uk)
- Centre de documentation de recherche et d'expérimentations sur les pollutions accidentelles des eaux (Cedre), France.
  Contact: Mr François Parthiot (Francois.Parthiot@cedre.fr)
- Coastal and Marine Resources Centre (CMRC), University College Cork, National University of Ireland, Cork, Ireland.
  Contact: Mr Declan Dunne (d.dunne@ucc.ie)
- Plymouth Marine Laboratory (PML), United Kingdom.
  Contact: Mr Steve Groom (sbg@pml.ac.uk)
- Institut français de recherche pour l'exploitation de la mer (Ifremer), France
  Contact: Mr Mickael Treguer (mickael.treguer@ifremer.fr)
- Norwegian Meteorological Institute (METNO), Norway.
  Contact: Mr Jan Ivar Pladsen (janip@met.no)

Author(s)

- Yassine Lassoued, UCCNUIC (CMRC), y.lassoued@ucc.ie
- Torill Hamre, NERSC, torill.hamre@nersc.no
- Roy Lowry, BODC, rkl@bodc.ac.uk
- Peter Walker, PML, petwa@pml.ac.uk
- Mikael Treguer, Ifremer, mickael.treguer@ifremer.fr
- François Parthiot, Cedre, Francois.Parthiot@cedre.fr

Document approval

- Document status: Release 1
- WP leader approval: 2010-06-09
- Quality Manager approval: 2010-06-09
- Coordinator approval: 2010-06-09

Revision History

Issue	Date	Change records	Author(s)
Draft	2010-04-23	First draft of the report.	Y. Lassoued
Draft	2010-06-03	Revised draft of the report incorporating feedback from Advisory Board and Steering Committee.	Y. Lassoued
1	2010-06-09	Final version of the report.	Y. Lassoued

Executive Summary

NETMAR aims to develop a pilot European Marine Information System (EUMIS) for searching, downloading and integrating satellite, in situ and model data from ocean and coastal areas. EUMIS will use a semantic framework coupled with ontologies for identifying and accessing distributed data, such as near-real time, forecast and historical data. This report identifies and assesses semantic web standards, semantic frameworks, and technologies that may be used as basis or building blocks for the NETMAR semantic framework. This report also identifies the semantics requirements for the NETMAR case studies, which will feed into the semantic framework design and implementation.

The review of semantic web standards focused on the most common World Wide Web Consortium (W3C) standards appropriate to the NETMAR project: Resource Description Framework (RDF), RDF Schema (RDFS) and Web Ontology Language (OWL) languages, the Simple Knowledge Organisation System (SKOS) data model and the SPARQL Protocol and RDF Query Language (SPARQL) query language. The main recommendation is that the NETMAR ontologies (but not thesauri or bridging mappings between semantic resources) shall be defined and delivered in a standard ontology language such as RDF/RDFS or OWL (Light, DL or Full), OWL 2 DL being the preferred candidate given its expressiveness and computability. An appropriate profile of OWL 2 (i.e. EL, QL or RL) should be carefully selected depending on the size and type of ontologies and applications to be developed. The SKOS model should be used where appropriate for thesauri or to represent semantic relationships among concepts of the NETMAR ontologies and possibly relationships with external ontologies.

The NETMAR ontologies may be stored in a relational database, and if needed be delivered to users through an ontology server using SPARQL. Alternatively, and RDF store such as Jena or Sesame may be used. Independently from these choices, we recommend the use of either Jena or Sesame as an Application Programming Interface (API) for accessing and querying the NETMAR ontologies. Jena is the preferred option. Also, we recommend the implementation of a custom NETMAR ontology browser that may reuse components of the jOWL JavaScript and/or standard Java Script, Adobe Flex or JavaFX graph libraries.

The main semantic frameworks for environmental information systems (EIS) reviewed in this report are: the International Coastal Atlas Network (ICAN) semantic mediation prototype, the Open Access Ontology/Terminology for the GMES Space Component (OTEG) semantic framework, the ORCHESTRA Semantic Catalogue Architecture, the NERC Data Grid Vocabulary Server, and the NASA Global Change Master Directory (GCMD). These semantic frameworks offer different architectures, capabilities and functionalities of interest to the NETMAR project.

The main requirements for the NETMAR semantic framework as identified through the defined NETMAR case studies are the following:

- *Data Discovery* – ability to perform semantic data discovery,
- *Service Discovery* – ability to perform semantic service discovery,
- *Interoperability* – ability to perform semantic data, metadata or service interoperability,
- *Service Chaining* – ability to perform semantics-based service chaining,
- *Multi-Domain Support* – support for multi-domain ontologies,
- *Multilingual Support* – support for multi-lingual vocabularies or ontologies,
- *Multi-Facet browsing/search* – support for multi-facet data/service search or browsing,
- *Ontology Browsing* – support for ontology or vocabulary browsing by users,
- *Smart Search* – support for meaning-based free text search by users,
- *Semantic Queries* – support for semantics-based queries by users.

None of the reviewed semantic frameworks offer all of these capabilities and functionalities on its own. Therefore, we recommend that the NETMAR semantic framework should build on existing semantic framework architectures and adapt these as needed to fulfil the NETMAR use case requirements. This allows the NETMAR project to draw upon best practices and proven solutions for semantic framework development in order to obtain a flexible and maintainable semantic framework for EIS. The detailed findings of this review, formulated as a set of development rules, recommendations and permissions (options), will be used as basis for the design and implementation of the NETMAR semantic framework.

Contents

1 Introduction

1.1 Background

NETMAR[1] aims to develop a pilot European Marine Information System (EUMIS) for searching, downloading and integrating satellite, in situ and model data from ocean and coastal areas. It will be a user-configurable system offering flexible service discovery, access and chaining facilities using OGC, OPeNDAP and W3C standards. It will use a semantic framework coupled with ontologies for identifying and accessing distributed data, such as near-real time, forecast and historical data. EUMIS will also enable further processing of such data to generate composite products and statistics suitable for decision-making in diverse marine application domains. Figure 1.1 illustrates how observations, derived parameters and predictions are retrieved from a distributed service network through standard protocols, and delivered through the EUMIS portal using ontologies and semantic frameworks to select suitable products and where new products can be generated dynamically using chained processing services.

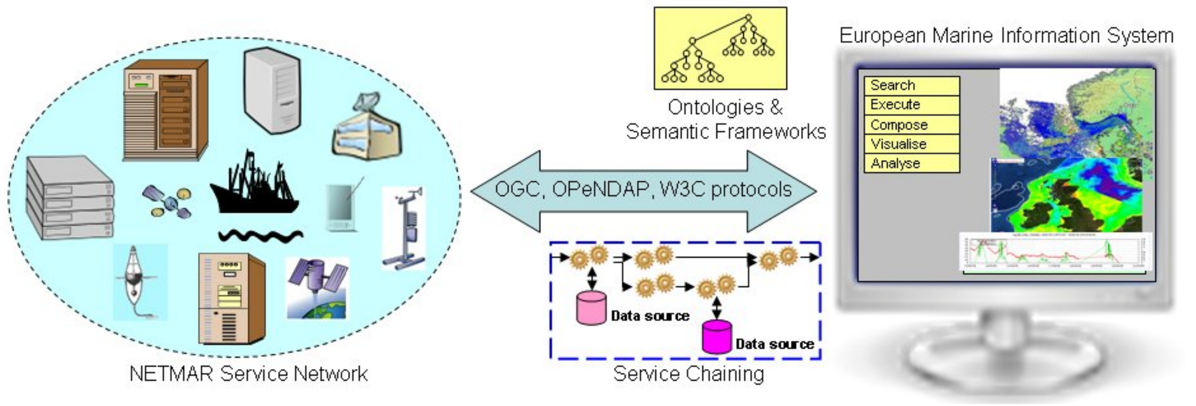


Figure 1.1. The NETMAR Service Network

1.2 Objective of this Report

This report identifies and assesses semantic frameworks, semantic web services standards, and technologies which may be used as basis or building blocks for the NETMAR semantic framework.

In addition, the report identifies the semantics requirements for the NETMAR case studies, which will feed into the semantic framework design and implementation.

This report does not provide an exhaustive list of semantic frameworks and standards. Rather it focuses on a set of pre-selected semantic frameworks and standards that have been identified as relevant to the NETMAR project. For a more complete list of semantic web technologies, standards and tools, readers can refer to the W3C Semantic Web Standards Wiki (SWSWiki)[2].

1.3 Terminology

This document uses several keywords that are defined as follows.

[1] http://www.netmar-project.eu/

[2] http://www.w3.org/2001/sw/wiki/Main_Page

Semantic Framework

The key concept in this report is “semantic framework”. In reality, there is no common definition for such a concept. Therefore it is important that we define what we mean by “semantic framework”.

The term “semantic framework” as used in this report means: a collection of classes, libraries, APIs, or applications that can be used to build semantics-aware information systems that integrate, manage, handle or deliver semantic knowledge related to the information system’s data and services.

Rule

Rules SHALL be followed to ensure compatibility and/or conformance with standards, directives or the project objectives. A rule is characterised by the use of the words SHALL and SHALL NOT.

Rules related to the NETMAR semantic framework are identified using keys in the format: RUL.SF.*X* (where *X* is a number).

Recommendation

Recommendations consist of advice to implementers that will affect the usability of the final module (here the NETMAR semantic framework). A recommendation is characterised by the use of the words SHOULD and SHOULD NOT.

Recommendations related to the NETMAR semantic framework are identified using keys in the format: REC.SF.*X* (where *X* is a number).

Permission

Permissions clarify areas of the specification that are not specifically prohibited. Permissions reassure the reader that a certain approach is acceptable and will cause no problem. Permissions are characterised by the use of the word MAY.

Permissions related to the NETMAR semantic framework are identified using keys in the format: PER.SF.*X* (where *X* is a number).

SHALL

“SHALL” is a keyword indicating a mandatory requirement. Designers SHALL implement such mandatory requirements in order to ensure conformance with the project objectives. This word is usually associated with a rule.

SHOULD

“SHOULD” is a keyword indicating flexibility of choice with a strongly preferred implementation. This word is usually associated with a recommendation.

MAY

“MAY” is a keyword indicating flexibility of choice with no implied preference. This word is usually associated with permissions.

IF – THEN – [ELSE]

Conditional rules, recommendations or Permissions MAY be used in the case where different scenarios are possible. In such a case we use an expression IF – THEN – [ELSE] (ELSE being optional).

1.4 Organisation of this Report

This report is organised as follows. In Chapter 2, we introduce W3C semantic web standards that that should be considered in the NETMAR semantic framework architecture.

In Chapter 3, we review generic open source ontology frameworks and APIs that may be used as basis or building blocks for the NETMAR semantic framework.

In Chapter 4, we introduce EIS semantic frameworks designed or developed as part of previous and ongoing projects. The introduced semantic frameworks are compared using a set of semantics capabilities and user interface functionalities.

Chapter 5 is dedicated to the identification of semantics requirements for the NETMAR case studies.

Recommendations resulting from chapters 4 and 5 are made in Chapter 6.

2 Semantic Web Standards

The Web Ontology Working Group of W3C developed several semantic web standards, including ontology languages, models and protocols. In this chapter, we introduce the main W3C semantic web standards. This report does not intend to provide a complete list of these standards. Rather it focuses on the ones recommended for use to build the NETMAR semantic framework as part of the NETMAR report D3.2, entitled "Review of available ontology tooling" [BODC10]. Also, a detailed technical description of these standards is beyond the scope of this report. Rather, we provide here high-level definitions of these standards and focus on how these standards should be used in practice and how they fit with each other. More details on these standards are to be found on the W3C website[3].

The standards dealt with in this chapter are split into two groups: ontology languages and tools, and ontology query languages.

2.1 Ontology Languages and Models

Report D3.2, entitled "Review of available ontology tooling" [BODC10], recommends the use of the RDF family of ontology languages, which consists of RDF, RDFS and OWL. D3.2 also recommends the use of the SKOS model (Simple Knowledge Organisation System). Subsections 2.1.1 to 2.1.4 introduce each of these languages and models using illustrations and show how they relate to each other and how they should be used in practice. This leads to a set of recommendations summarised in subsection 2.1.5.

2.1.1 *Resource Description Framework (RDF)*

The Resource Description Framework (RDF) [RDF04] is a general-purpose model for representing information in the Web. RDF was designed to describe resources on the web by means of *statements*. A *statement* in RDF describes a *resource* using *properties* and *property values*. A *resource* is a web page or a real world object such as an *organisation* or a *project*. A *property* is a relationship that associates with a resource some "value". For example *name* and *start date* are properties. The associated value is called *property value*, e.g., "NETMAR" or "01 February 2010". In RDF, values may be atomic in nature (text strings, numbers, etc.) or other resources, which in turn may have their own properties. A collection of these properties that refers to the same resource is called a *description*.

An RDF statement is defined as a triple *(subject, predicate, object)*. The *subject* refers to the resource being described, the *predicate* to the property and the *object* to the property value. For example *(netmar_project, name, "NETMAR")* is a statement.

Basically, RDF provides a syntax-independent model for representing resources and descriptions. An RDF description can be represented as a directed labelled graph. The so-called *RDF graph* has nodes and labelled directed arcs that link pairs of nodes. It is represented as a set of RDF triples, where each triple contains a subject node, predicate and object node. Nodes are RDF URI (Uniform Resource Identifier) references, RDF literals or blank nodes. Blank nodes may be given a document-local, non-RDF URI references identifier called a blank node identifier. Predicates are RDF URI references and can be interpreted as either a relationship between the two nodes or as defining an attribute value (object node) for some subject node.

[3] http://www.w3.org/

An example of an RDF graph is illustrated in Figure 2.1, which represents the following statements:

- The *collection date* of Seabed Sample 1 is 08 July 2007,
- The *classification* of Seabed Sample 1 is Folk Class 10,
- The *name* for Folk Class 10 is "Sandy gravel",
- The *code* for Folk Class 10 is “sG”.

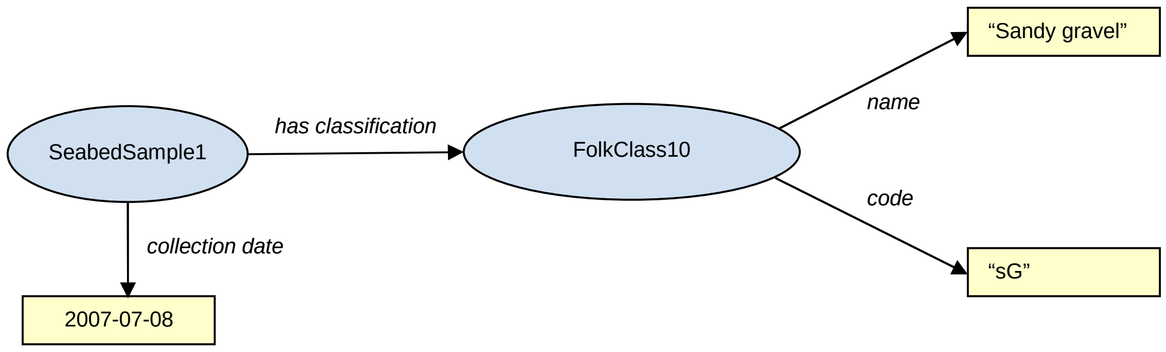


Figure 2.1. Illustrative Example of a Resource Description for a Seabed Sample Resource

An XML syntax for RDF graphs, called RDF/XML [BM04], has been developed by W3C. The purpose of it is to provide a common machine processable and extensible encoding for RDF graphs. In RDF/XML, nodes and predicates are represented in XML terms: element names, attribute names, element contents and attribute values. For instance, an RDF/XML encoding of the previous seabed sample example would be something similar to the following:

```
<rdf:Description rdf:about="SeabedSample1">
      <ex:folkClassification>
            <rdf:Description rdf:about="FolkClass10">
                  <ex:name>Sandy gravel</ex:name>
                  <ex:code>sG</ex:code>
            </rdf:Description>
      </ex:folkClassification>
      <ex:collectionDate>2007-07-08</ex:collectionDate>
</rdf:Description>
```

Another textual syntax for RDF graphs, also specified by W3C, is called Turtle (Terse RDF Triple Language) [BB08]. Turtle allows RDF graphs to be completely written in a compact and natural text form, with abbreviations for common usage patterns and data types. Turtle provides levels of compatibility with the triple pattern syntax of the SPARQL W3C Recommendation (c.f. Section 2.5).

2.1.2 *RDF Schema (RDFS)*

As explained in the previous subsection, RDF provides a way to express statements about resources. However, RDF user communities also need the ability to define the vocabularies (terms) they intend to use in those statements, specifically, to indicate that they are describing specific kinds or classes of resources, and will use specific properties in describing those resources. For example, an organisation example.com would want to describe classes such as *ex:SeabedSample* and *ex:FolkClass*, and use properties such as *ex:folkClassification*, *ex:collectionDate*, *ex:name* and *ex:code* to describe them. RDF itself provides no means for defining such application-specific classes and properties. Instead, such classes and properties are described as an RDF vocabulary, using extensions to RDF provided by the RDF Vocabulary Description Language, referred to as RDF Schema (RDF-S or RDFS) [BG04]. RDF Schema further extends RDF by adding more modelling primitives often found in ontology languages like classes, class inheritance, property inheritance, domain, range restriction.

The following example shows an RDF schema for seabed samples. It defines the *SeabedSample* and *FolkClass* classes, and the various properties *name*, *code*, and *hasClassification*.

```
<rdf:Description rdf:ID="SeabedSample">
  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
</rdf:Description>
<rdf:Description rdf:ID="FolkClass">
  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
</rdf:Description>
<rdf:Description ID="name">
  <rdf:type resource="http://www.w3.org/2000/01/rdf-schema#Property"/>
  <rdfs:domain rdf:resource="#SeabedSample"/>
  <rdfs:range rdf:resource="http://www.w3.org/TR/xmlschema-2/#string"/>
</rdf:Description>
<rdf:Description ID="code">
  <rdf:type resource="http://www.w3.org/2000/01/rdf-schema#Property"/>
  <rdfs:domain rdf:resource="#SeabedSample"/>
  <rdfs:range rdf:resource="http://www.w3.org/TR/xmlschema-2/#string"/>
</rdf:Description>
<rdf:Description rdf:ID="hasClassification">
  <rdf:type resource="http://www.w3.org/2000/01/rdf-schema#Property"/>
  <rdfs:domain rdf:resource="#SeabedSample"/>
  <rdfs:range rdf:resource="#FolkClass"/>
</rdf:Description>
```

RDFS defines a set of very useful annotation properties, such as *label*, *comment*, *isDefinedBy* and *seeAlso*, with multilingual support. For instance one may associate several labels and comments with an entity (class, instance, property, etc.) using different languages. For instance, one can define three labels for the concept “Gravel” in three languages (English, French and German) as shown in the following example.

```
ex:Gravel rdfs:label "Gravel"@en
ex:Gravel rdfs:label "Gravier"@fr
ex:Gravel rdfs:label "Kies"@de
```

2.1.3 *OWL*

OWL [MH04, W3C09], the Web Ontology Language is actually a family of ontology languages built on top of RDF. OWL has more facilities for expressing meaning and semantics than RDF, and RDF-S, and thus OWL goes beyond these languages in its ability to represent machine interpretable content on the Web. OWL adds more vocabulary for describing properties and classes: among others, relations between classes (e.g. disjointness), cardinality (e.g. "exactly one"), equality, richer typing of properties, characteristics of properties (e.g. symmetry, transitiveness), and enumerated classes.

OWL became a W3C recommendation in 2004 [MH04]. We refer to the 2004 version of OWL as OWL 1. In 2007, the W3C OWL Working Group started extending OWL in order to produce a new W3C recommendation for an updated OWL. The so-called OWL 2 [W3C09] became a W3C recommendation in November 2009.

2.1.3.1 OWL 1

OWL 1 provides three increasingly expressive sublanguages: OWL Lite, OWL DL and OWL Full, which are introduced below.

- OWL Lite supports those users primarily needing a classification hierarchy and simple constraints. For example, while it supports cardinality constraints, it only permits cardinality values of 0 or 1. It should be simpler to provide tool support for OWL Lite than for its more expressive relatives. OWL Lite also has a lower formal complexity than OWL DL, see the section on OWL Lite in the OWL Reference [MH04] for further details.
- OWL DL supports those users who want the maximum expressiveness while retaining computational completeness (all conclusions are guaranteed to be computable) and decidability (all computations will finish in finite time). OWL DL includes all OWL language constructs, but they can be used only under certain restrictions (for example, while a class may be a subclass of many classes, a class cannot be an instance of another class).
- OWL Full is meant for users who want maximum expressiveness and the syntactic freedom of RDF with no computational guarantees. For example, in OWL Full a class can be treated simultaneously as a collection of individuals and as an individual in its own right. It is unlikely that any reasoning software will be able to support complete reasoning for every feature of OWL Full.

2.1.3.2 OWL 2

OWL 2 [W3C09] extends OWL 1 and inherits the language features and design decisions of OWL 1. It adds several new features to OWL 1, including:

1. Syntactic sugar, i.e. syntaxes designed to make common statements easier to express, e.g., constructs such as `DisjointUnion` to express that a class is the disjoint union of other classes, and `DisjointClasses` to express that a set of classes are disjoint;
2. New constructs for increasing the expressiveness of properties; for instance one can define a subclass of objects that are related to themselves by a given property (e.g., a self regulating process is a process that regulates itself), one also can define reflexive, irreflexive and asymmetric object properties, disjoint properties, properties composition (i.e., a property that is a composition of other properties such as *grand-parent = father of father*), and keys (i.e., unique identifiers of objects);
3. Extended data type capabilities such as support for extra data types (e.g., various kinds of numbers, XML schema data types, time instants, etc.), support for data type restrictions (e.g., sting minimum and maximum lengths, minimum and maximum values for numbers, etc.), constructs for defining new data types, and data range combination using the intersection, union and complementary constructs;
4. Simple meta-modelling capabilities, such as the possibility of using the same term for both a class and an instance;
5. Extended annotation capabilities, such as the ability to annotate ontologies, entities, anonymous individuals, axioms and even annotations themselves;
6. Other innovations and minor features including declarations (the possibility of declaring entities as part of the vocabulary of an ontology), top and bottom properties (analogous to the to and bottom classes: Thing and Nothing), possibilities of using IRIs (Internationalized Resource Identifiers) rather than URIs, and improved management of imports and versioning.

OWL 2 is fully backwards compatible, which means that any OWL 1 ontology remains a valid OWL 2 ontology, with identical inferences.

OWL 2 has three profiles (i.e., sub-languages) that offer various advantages in particular application scenarios: OWL 2 EL, OWL 2 QL and OWL 2 RL [MGH09].

- OWL 2 EL is suitable for applications using large ontologies, i.e. containing a large number of classes or properties. It captures the expressive power used by many of such ontologies. The basic reasoning problems for OWL 2 DL ontologies can be performed in polynomial time with respect to the size of the ontology.
- OWL 2 QL is suitable for applications using very large volumes of instance data and where query answering is the main reasoning task. It is designed in such a way that query answering can be performed in LOGSPACE with respect to the size of data. As in OWL 2 QL, reasoning can be answered in polynomial time with respect to the ontology size. OWL 2 QL is aimed at applications where data are stored in relational database management systems (RDBMS) and queried through an OWL 2 QL ontology. Only a simple query rewriter is required to rewrite ontology queries into SQL queries that can be answered by the RDBMS. However, the expressive power of OWL 2 QL is quite limited.
- OWL 2 RL is aimed at applications requiring scalable reasoning without sacrificing too much expressive power. Rule-based reasoning engines can be used for reasoning on OWL 2 RL ontologies. The basic reasoning problems for OWL 2 RL ontologies can be answered in polynomial time with respect to the size of the ontology.

Specifications of the various OWL 2 profiles and details of their computational requirements can be found in [MGH09]. However, it is important to mention here that OWL 2 still have the sub-languages OWL 2 Lite, DL and Full. Each of the OWL 2 profiles (EL, QL and RL) is more restrictive than OWL 2 DL.

2.1.4 *Simple Knowledge Organisation System (SKOS)*

SKOS [MB09] is a common data model for knowledge organisation systems such as thesauri, classification schemes, taxonomies, subject-heading systems, and taxonomies within the framework of the Semantic Web. As of August 2009, the SKOS Specifications [MB09] are published as W3C recommendations.

SKOS provides a standard way to represent knowledge organisation systems using the Resource Description Framework (RDF). Encoding this information in RDF allows it to be passed between computer applications in an interoperable way.

Using RDF also allows knowledge organisation systems to be used in distributed, decentralised metadata applications. Decentralised metadata is becoming a typical scenario, where service providers want to add value to metadata harvested from multiple sources.

2.1.4.1 SKOS Data Model

The SKOS data model views a knowledge organisation system as a ***concept scheme*** comprising a set of ***concepts***. SKOS concept schemes and concepts are identified by URIs. SKOS concepts can be:

- Labelled with any number of lexical (UNICODE) strings in any given natural language, such English or Japanese; one of these given labels in any given language can be indicated as the “preferred” label for that language,
- Assigned one or more notations, which are lexical codes used to uniquely identify the concept within the scope of a given concept scheme,
- Documented with notes of various types (scope notes, definitions, editorial notes, etc.),
- Linked to other SKOS concepts via semantic relation properties, thus giving support for hierarchical and associative links between SKOS concepts,
- Grouped into collections which can be labelled and/or ordered,
- Mapped to other SKOS concepts in different concept schemes, using four basic types

of mapping links: hierarchical, associative, close equivalent and exact equivalent.

An example of SKOS data expressed as an RDF graph is illustrated below. The graph is expressed in Turtle (c.f. Section 2.1).

```
<A> rdf:type skos:Concept ;
    skos:prefLabel "Gravel"@en ;
    skos:altLabel "G"@en ;
    skos:broader <B> ;
    skos:inScheme <S> .

<B> rdf:type skos:Concept ;
    skos:prefLabel "Folk Class"@en ;
    skos:altLabel "Folk Sediment Class"@en ;
    skos:topConceptOf <S> .

<S> rdf:type skos:ConceptScheme ;
    dct:title "Marine Geology Thesaurus" ;
    skos:hasTopConcept <B> .
```

In this example, it is expressed that *A* is a concept called, in English, “Gravel” (preferred label) or alternatively “G”. Concept *A* has a broader concept *B*, called “Folk Sediment Class” or preferably “Folk Class”. Both concepts *A* and *B* belong to a concept scheme *S* called “Marine Geology Thesaurus”, concept *B* being a top level concept of this scheme, i.e. is not narrower than any other concept in *S*.

Like RDFS annotations, SKOS annotations are multilingual. Therefore, one can associate with a concept several labels and definitions in different languages.

2.1.4.2 Usage

SKOS may be used on its own, or in combination with formal knowledge representation languages such as the Web Ontology language (OWL) to express and exchange knowledge about a domain. SKOS itself is defined as an OWL ontology in which labelling and documentation properties are implemented as OWL annotation properties, and semantic relationships are implemented as object properties.

2.1.5 *Discussion*

2.1.5.1 RDF vs. OWL

OWL and RDF are “pretty much the same thing.” In fact, OWL is built on top of RDF. However, OWL was designed to be higher-level a language and to offer greater machine interpretability than RDF. In addition, various OWL profiles exist, which are tailored for specific application types in order to provide them with the required levels of expressiveness and computability. OWL is also part of the Semantic Web vision. For these reasons, we strongly recommend OWL for developing the NETMAR ontologies.

2.1.5.2 OWL 1 vs. OWL 2

As mentioned earlier (c.f. subsection 2.1.3.2), OWL 2 is fully backwards compatible. Therefore, any valid OWL 1 ontology remains a valid OWL 2 ontology, with identical inferences. OWL 2 only extends OWL 1 with new constructs and syntactic sugar. The use of the OWL 2 features in building the NETMAR ontologies will depend on whether they are required or not.

2.1.5.3 SKOS in OWL

The SKOS reference [MB09] defines the SKOS concept scheme (`skos:ConceptScheme`) and concept (`skos:Concept`) as OWL classes (`owl:Class`). Therefore, any concept scheme should be an instance (individual) of `skos:ConceptScheme` and any concept should be an instance of `skos:Concept`. For example, one may define a concept scheme called *Place*. Such a concept scheme should be an instance of `skos:ConceptScheme`. The concepts *European Country* and *Ireland* (which belong to the *Place* concept scheme) should be defined as an instances of `skos:Concept`. Semantic relationships are defined as object properties. For instance one may express that *Ireland* is narrower a term than *European Country*.

```
ex:Place rdf:type skos:ConceptScheme
ex:Ireland rdf:type skos:Concept
ex:EuropeanCountry rdf:type skos:Concept
ex:Ireland skos:inScheme ex:Place
ex:EuropeanCountry skos:inScheme ex:Place
ex:Ireland skos:broader ex:EuropeanCountry
```

One of the common questions often asked by ontology developers when using SKOS with OWL is: can a concept or a concept scheme such as European Country or Place be treated as a class in its own right. The reason is: SKOS concepts or concept schemes can be seen as meta-classes, i.e., their instances can be any concepts (classes or instances) occurring in a vocabulary or an ontology. For instance Place can be seen as a class of objects that share common properties such as geographic location, code, name, etc. In the same way European Country can be seen as a class of objects having more specific properties such as year of EU entry, etc. Therefore, one may need to define class-level characteristics of concepts or concept schemes.

The SKOS primer [IS09] states that "*SKOS does not take a stance with respect to the flavour of OWL – OWL Full or OWL DL – to be used together with SKOS.*" While OWL Full allows its users to handle this situation by treating classes as objects, OWL DL prevents its users from doing so. Nevertheless, workarounds exist to handle this situation. For instance, one may define two entities for places: an OWL class called PlaceClass and a SKOS concept scheme called PlaceScheme.

```
ex:PlaceClass rdf:type owl:Class
ex:PlaceScheme rdf:type skos:ConceptScheme
```

One can then explicitly link them either using a restriction on PlaceClass (such as all PlaceClass instances necessarily and sufficiently belong to concept scheme PlaceScheme) or through an owl:annotation such as:

```
ex:PlaceClass rdf:correspondingScheme ex:PlaceScheme
```

The same mechanism could be applied to the European Country class and concept.

Note that in OWL 2, one can use the same name for an instance and a class, but still needs to link them as described above.

2.1.5.4 SKOS Annotations vs. RDFS Annotations

RDFS defines the *label* annotation that can be used to attach labels to entities (classes, instances, properties, etc.). SKOS, too, defines its own set of labelling annotations (preferred label, alternative label and hidden label). When developing OWL ontologies using the SKOS

model one may ask: “Which labels should one use: SKOS labels, or RDFS labels, or both?”
It turns out that all the SKOS labels are sub-properties of the RDFS label (`rdfs:label`). Therefore it is sufficient to use the SKOS labels. A reasoning engine should be ale to conclude that any SKSOS label attached to a concept is also an RDFS label.

2.2 Ontology Query Languages

2.2.1 *SPARQL*

SPARQL [PS08] is a query language for getting information from RDF graphs (c.f. section 2.1). It provides facilities to:

- Extract information in the form of Unique Resource Identifiers (URIs), blank nodes and literals,
- Extract RDF sub-graphs,
- Construct new RDF graphs based on information in the queried graphs.

The SPARQL query language is based on matching graph patterns. The simplest graph pattern is the triple pattern, which is like an RDF triple, but with the possibility of a variable instead of an RDF term in the subject, predicate or object positions. Combining triple patterns gives a basic graph pattern, where an exact match to a graph is needed to fulfil a pattern. For instance, consider the painting example of section 3.3. More particularly, consider the following RDF triple that expresses that the *name* of Folk Class 10 is "Sandy gravel":

```
<http://example.org/geology/FolkClass10> <http://example.org/geology/name> "Sandy gravel"
```

In this triple, Folk Class 10 and the *name* property have been identified by their URIs (both in the format http://example.org/geology/*).

The example below shows a SPARQL query to find the name for a Folk Class from the information in the given RDF graph.

```
SELECT ?name
WHERE
{
   <http://example.org/geology/FolkClass10> <http://example.org/geology/name> ?name
}
```

The query consists of two parts, the Select clause and the Where clause. The Select clause identifies the variables to appear in the query results, and the Where clause has one triple pattern.
This query, on the data above, has one solution: "Sandy gravel".

2.2.2 *GeoSPARQL*

A draft specification of a geographic query language for RDF, called GeoSPARQL, has recently been proposed as an OGC draft candidate standard [PH10]. The idea is (i) to develop a feature model for geographic data to be expressed as RDF triples, and (ii) to extend SPARQL by adding support for handling geometries and for supporting and evaluating spatial predicates.

It is very important to note that the GeoSPQRQL specification is not an OGC recommendation yet. It is still work in progress and is subject to change without notice.

However, the NETMAR community should keep an eye on the development of this standard, and, if possible, contribute to it.

GeoSPARQL and its feature model may be used in NETMAR for representing and querying metadata place keywords, which may be defined as concepts with geographic properties (location, bounding box, etc.).

2.3 Recommendations

The W3C semantic web standards aim to pave a common ground for semantic web applications to interoperate and participate in building the semantic web. The standards introduced above are the most common ones and the most relevant to the NETMAR project. The NETMAR semantic framework architecture should therefore comply with these standards in order for it to be interoperable with other standards-based semantic frameworks and applications. More specifically, the following recommendations should be considered when designing and implementing the NETMAR semantic framework.

RUL.SF.1

The NETMAR ontologies (but not thesauri or bridging mappings between semantic resources) SHALL be defined using a standard ontology language such as RDF/RDFS or OWL in order to facilitate interoperability with external standards-based semantic frameworks and ontologies.

REC.SF.1

Compared to RDF and RDFS, OWL offers more expressiveness and machine interpretability and allows for rich semantic relationships and rules. Therefore, the NETMAR ontologies SHOULD be implemented in OWL rather than RDF.

REC.SF.2

IF OWL is to be used as the ontology language for the NETMAR semantic framework, THEN, in order to cope with the evolution of the W3C Web Ontology Language and to profit from the new features of OWL 2, the NETMAR ontologies SHOULD be defined in OWL 2 rather than OWL 1. It is important to note here that most popular ontology tools and software already started supporting OWL 2.

RUL.SF.2

IF OWL is to be used as the ontology language for the NETMAR semantic framework, THEN OWL DL SHALL be used rather than OWL Lite or OWL Full. This would guarantee maximum expressiveness while retaining computational completeness and decidability.

REC.SF.3

IF, following REC.SF.1, REC.SF.1, and RUL.SF.2, OWL 2 DL is to be used as the ontology language in NETMAR, THEN an appropriate profile (i.e., OWL 2 EL, OWL 2 QL, or OWL 2 RL) SHOULD be carefully chosen depending on the type of application to be implemented. The use of OWL 2 profiles will ensure that reasoning be done in polynomial time. Moreover, the choice of the appropriate profile will ensure optimum balance of expressiveness vs. performance.

REC.SF.4

The SKOS model SHOULD be used for encoding thesauri and as a mechanism for specifying semantic relationships among concepts of the NETMAR ontologies and possibly relationships with external ontologies (if required). This would ensure that semantic relationships be implemented in a standard way in accordance with the W3C recommendations, which would facilitate interoperability with other ontologies and semantic frameworks.

RUL.SF.3

IF, according to REC.SF.4, SKOS is to be used in the NETMAR ontologies, THEN, according to the OWL DL restrictions, concepts and concepts schemes SHALL (i.e., must) be implemented only as instances of skos:Concept and skos:ConceptSchemes and not as owl:Class.

PER.SF.1

IF, when using SKOS with OWL DL, any concepts or concept schemes need to be defined as classes, THEN a workaround MAY be used. For instance one MAY define a separate class and an instance corresponding to the same concept or concept scheme, and link them using an owl:annotation or a restriction on the class instances.

REC.SF.5

XML is commonly accepted as the data format for sharing information on the web. Therefore, RDF/XML SHOULD be used as an exchange format for the NETMAR ontologies rather than Turtle.

PER.SF.2

The NETMAR ontologies MAY be queried, internally, by the NETMAR semantic framework using the SPARQL protocol.

REC.SF.6

IF the NETMAR ontologies are to be made publicly available THEN they SHOULD be delivered through an ontology server using a standard protocol such as SPARQL. This would allow external users to perform semantic queries on the NETMAR ontologies in a standard way and to reuse these in their applications.

REC.SF.7

Although GeoSPARQL is not yet a recommendation, the NETMAR community SHOULD keep an eye on the progress of this work and SHOULD consider actively participating in the development of this standard.

PER.SF.3

GeoSPARQL and its feature model MAY be used in NETMAR for representing and querying metadata place keywords, which MAY be enriched using geospatial properties such as geographic location or bounding box information.

3 Semantic Frameworks

A variety of semantic frameworks are available online. The W3C Semantic Web Standards Wiki (SWSWiki)[4] provides a long list of references to such frameworks, technologies and tools. In this chapter we introduce candidate open source standards-compliant semantic frameworks and technologies for the NETMAR system implementation.

3.1 Semantic Frameworks

Semantic frameworks generally cover one or more of the following:

- Application programming interfaces (API) for managing, handling, and querying ontologies,
- RDF stores, i.e. systems for storing and managing RDF triples,
- Ontology inference engines (reasoners).

In this section we introduce the most popular open source semantic frameworks that could be used to build the NETMAR semantic framework. The reviewed semantic frameworks may provide one or more of the features listed above. Therefore, it is very difficult to classify them by type. More semantic frameworks and tools are identified and introduced in [BODC10]. We only consider here the ones recommended in [BODC10].

3.1.1 *Jena and Jena 2*

Jena[5] is an open source Java framework for building semantic web applications. It provides an API that supports RDF, RDFS, OWL and SPARQL and includes a rule-based inference engine. The Jena Framework includes:

- An RDF API,
- Reading and writing RDF in RDF/XML,
- An OWL API,
- In-memory and persistent storage,
- Database support,
- SPARQL query engine.

The Jena2[6] inference subsystem is designed to allow a range of inference engines or reasoners to be plugged into Jena. Such inference engines are used to derive additional RDF assertions from instance data and class descriptions using the axioms and rules associated with the reasoner. Jena2 is designed to be quite general and, in particular, it includes a generic rule engine that can be used for many RDF processing or transformation tasks.

Jena and Jena2 have good quality documentation and tutorials, which makes them quite easy to use. In addition they are still supported and frequently updated.

Jena per se does not support OWL 2 yet. However, some OWL 2-compliant reasoners that have Jena interfaces, such as Pellet[7], do support OWL 2, and therefore allow Jena users to handle OWL 2 ontologies.

[4] http://www.w3.org/2001/sw/wiki/Main_Page

[5] http://jena.sourceforge.net/

[6] http://jena.sourceforge.net/inference/

[7] http://clarkparsia.com/pellet/

3.1.2 *Sesame*

Sesame[8] is an open source RDF framework with support for RDF Schema inferencing and querying. It supports the SPARQL and SeRQL query languages. The framework is fully extensible and configurable with respect to storage mechanisms, inferencers, RDF file formats, query result formats and query languages. Sesame is written in Java and is platform independent. It provides a data storage solution, a JDBC-like user API, and a ReST-ful HTTP interface supporting the SPARQL Protocol for RDF.

Sesame is primarily intended to store, manage and query RDF triples. As OWL is built on top of RDF, Sesame handles OWL ontologies as RDF triples. However, it does not offer an OWL view of them.

The Sesame API has very good quality documentation and, like Jena, is continuously supported and updated.

3.1.3 *OWL API*

The OWL API[9] is a Java API for handling OWL ontologies, primarily maintained by the University of Manchester. Its latest version is focused towards OWL 2. The API provides components for parsing and writing several ontology languages and syntaxes, including but not limited to, RDF/XML, OWL/XML, OWL Functional Syntax and Turtle. The OWL API is primarily targeted at representing OWL-DL. This does not mean that it does not handle OWL Full ontologies, but a number of design decisions reflect this assumption.

The OWL API provides reasoner interfaces for FaCT++[10], HermiT[11], Pellet (mentioned above in subsection 3.1.1) and Racer[12] however it does not provide in itself any implementations of reasoners. In addition, it does not provide ontology query interfaces and does not provide support for RDF triple stores (ontologies are loaded into memory only).

3.1.4 *Simple Ontology Framework API (SOFA)*

SOFA[13] is a simple but powerful ontology API that allows for inter-operation between different ontology description formats. SOFA is not tied down to a particular storage layer and can easily be integrated into any application that requires an ontology manager. Due to the structure of the API, virtually any Java object can be used to model ontology data type nodes, allowing the model to be as complex or simple as necessary. Features of the SOFA include:

- Multiple inheritance, allowing the discovery of nodes beyond the first set of sub, or super-concepts;
- Ontology inter-operation, so two ontologies in the same session can talk to each other and use the same resources;
- Inferencing and reasoning about relationships;
- Support for W3C OWL, RDF and RDF Schema;
- Ontology creation and querying.

[8] http://www.openrdf.org/

[9] http://owlapi.sourceforge.net/

[10] http://owl.man.ac.uk/factplusplus/

[11] http://hermit-reasoner.com/

[12] http://www.racer-systems.com/

[13] http://sofa.projects.semwebcentral.org/

The latest version of SOFA (version 0.3) was released in 2005, and the project, which ran from 2004 to 2005, does not seem to be updated or supported any more.

3.1.5 *RDF API for PHP (RAP)*

RAP[14] is a semantic web toolkit for PHP developers. RAP started as an open source project at the Freie Universität Berlin in 2002 and has been extended with internal and external code contributions since then. Its latest release includes:

- A statement-centric API for manipulating RDF graphs as a set of statements,
- A resource-centric API for manipulating RDF graphs as a set of resources,
- In-memory or database model storage,
- An inference engine supporting RDF-Schema reasoning and some OWL entailments,
- An RDF server,
- A graphical user-interface for managing database-backed RDF models,
- Drawing graph visualisations.

RAP offers two different programming interfaces for manipulating RDF graphs: The statement-centric Model API which allows you to manipulate an RDF graph as a set of statements; and the resource-centric ResModel API for manipulating an RDF graph as a set of resources.

The Model API supports adding, deleting, and replacing statements inside a model as well as adding entire models. StatementIterators allow sequential access to all statements within a model. RAP partially supports OWL.

The latest of version of RAP was released in February 2008 and does not seem to have been updated since.

3.1.6 *Comparison*

Currently, Jena and Sesame are the two most popular open source implementations for RDF store. Both of them have quite comparable easy to use and powerful APIs and can store and manage RDF triples in a relational database management system (RDBMS). Jena supports MySQL, PostgreSQL, Oracle, SQL Server, Derby and HSQLDB, whereas Sesame only supports MySQL and PostgreSQL. Jena, in addition provides an OWL view of OWL ontologies, which Sesame does not. However, Sesame provides an HTTP interface supporting the SPARQL protocol for RDF.

Regardless of the RDF database being used, either Jena or Sesame can be used independently to access the RDF store. Nevertheless, APIs exist for interoperating Jena and Sesame. The most popular ones are:

- The Jena Sesame Model project[15], which allows developers to access Sesame databases through Jena's model abstraction,
- The Sesame-Jena Adapter project[16], which provides access to Jena models through the Sesame API.

Unless really required, it is not recommended to access Jena with Sesame or Sesame with Jena as this would add another layer of complexity, and maintenance and update difficulties (what happens when one wants to upgrade Jena or Sesame or both?).

[14] http://www.seasr.org/wp-content/plugins/meandre/rdfapi-php/doc/tutorial/introductionToRAP.htm

[15] http://sites.google.com/site/wjfang2/jenasesamemodel

[16] http://sjadapter.sourceforge.net/

The OWL API has good quality and is very well documented and maintained with support for OWL 2. Nevertheless, unlike Jena and Sesame, it does not offer support for RDF stores or ontology query languages.

While Jena and Sesame offer low-level classes and methods for ontology management, SOFA offers a higher-level API, providing user-friendly and easy-to-use classes and methods. Unlike Jena and Sesame, SOFA is not tied to any particular ontology language or store. Rather it represents ontologies at a conceptual level, using an ontology model that is quite compatible with the OWL one. However.

Use of any of the above-mentioned frameworks versus that of RAP only depends on the programming language preference as the former are Java APIs and the latter is a PHP API. However, it is very important to note that RAP does not seem to have been maintained since February 2008.

3.1.7 *Recommendations*

REC.SF.8

Given that Jena and Sesame are the most established semantic frameworks and given the various features they provide (such as RDF stores, APIs, support for OWL and SPARQL) the NETMAR semantic framework SHOULD be developed in Java using Jena or Sesame.
More specifically, the authors recommend that Jena SHOULD be used rather than Sesame as the former provides an OWL view of ontologies, which is very handy when handling OWL ontologies.

REC.SF.9

Although Jena and Sesame can communicate with each other using the Jena Sesame Model or the Sesame-Jena adaptor, for sake of ease of implementation and maintenance, one SHOULD use the Jena API to access Jena databases and the Sesame API to access Sesame databases.

REC.SF.10

Scalability of the ontology base is an important issue that could be solved using RDF stores or relational databases. Therefore, the NETMAR ontologies SHOULD be stored and managed in an RDF store or a relational database.

3.2 Online Ontology Browsers

Report D3.2 [BODC10] introduces a set of web-based ontology browsers that could be used to browse the NETMAR ontologies in the NETMAR semantic framework and portal. Most of the available browsers are either still in development or old proof of concept projects that have not been taken any further. The only tool that has been tested and that is still maintained is jowl which is introduce in the next subsection.

3.2.1 *jOWL*

jOWL[17] is a jQuery[18] plugin for navigating and visualising OWL-RDFS documents. The current version of jOWL (1.0) is tested on Internet Explorer 7+, Firefox 3, Opera 0.95, and Safari. jOWL provides various visualisation components such as tree view and navigation bar. Advanced reasoning can be enabled or disabled. A demonstration screenshot is shown in Figure 3.1. The figure shows 4 components provided by jowl:

1. A navigation bar that allow users to browse the ontology classes by simple mouse clicks; when a class is clicked, its super-classes and sub-classes are loaded into the navigation bar;
2. A text search component, for searching ontology classes, with a list of suggestions from the ontology class names;
3. A direct individuals component for containing the individuals of a selected class;
4. A tree view component for hierarchical visualisation of the ontology classes.

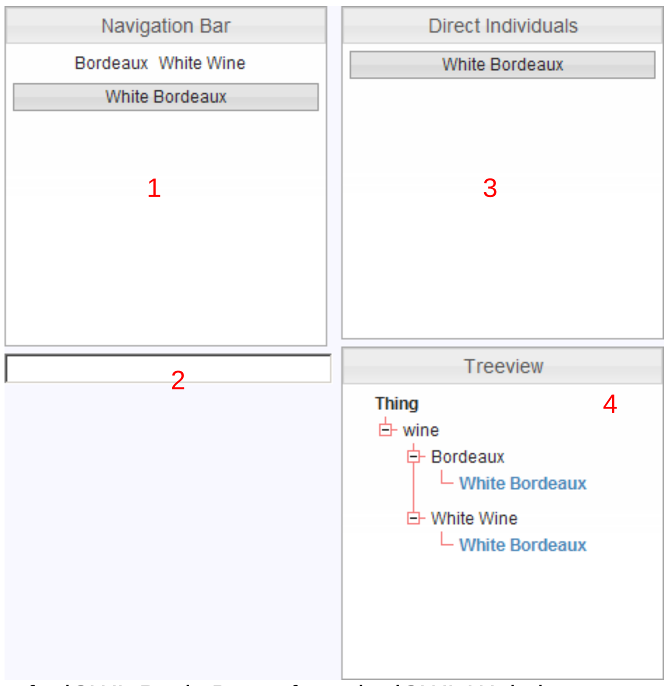


Figure 3.1. Screenshot of a jOWL Basic Demo from the jOWL Website

jOWL can be used for data loading and reasoning. Ontologies can be stored either locally or remotely. jOWL can be used on its own or in combination with other JavaScript libraries and components to build more advanced visualisations such as hyperbolic tree visualisations or fore-directed graphs. Good quality documentation is available for jOWL and quite a few demos are available from the jOWL website.

3.2.2 *Discussion*

The jOWL API introduced above is an API for building generic OWL ontology browsers, i.e. browsers that support any OWL ontology regardless of their content and structure. The ontology is loaded from a remote or local file and displayed using generic components such as the navigator bar, the tree view, etc. Although the use of such an API may be justified by the fact that the API is generic and provides predefined generic components, the

[17] http://jowl.ontologyonline.org/

[18] jQuery (http://jquery.com/) is JavaScript Library that simplifies HTML document traversing, event handling, animating, and Ajax interactions for rapid web development.

implementation of a custom NETMAR ontology browser is a preferred option for the reasons outlined below.

The jOWL API provides a set of generic components for visualising the class hierarchy and instances of OWL ontologies. The NETMAR ontologies will have a particular structure in which concepts will be represented as SKOS concepts (i.e. instance of `skos:Concept`) and linked to each other using SKOS semantic relationships. What is important to visualise then is not the ontology as a whole (class structure and instances). Rather, it is a set of graphs of instances (concepts) corresponding to various viewpoints of the ontologies, for instance the hierarchy of concepts according to a given semantic relationship such as `skos:narrower`, or the graph of concepts using the `skos:related` relationship. This requires a customised ontology browser that displays graphs of concepts built by the semantic framework rather than a generic ontology browser that displays OWL files.

3.2.3 *Recommendations*

REC.SF.11

The NETMAR semantic framework SHOULD implement a customised ontology browser for displaying useful concept graphs based on the SKOS semantic relationships. Examples of such graphs include the hierarchy of concepts using the `skos:narrower` and `skos:broader` relationships, and the graph of semantically related terms.

PER.SF.4

Whenever suitable, jOWL component MAY be adapted and reused to build the NETMAR ontology browser. Other standard graph libraries such as JavaScript, Adobe Flex or JavaFX graph libraries (trees, graphs, etc.) MAY also be used to build the NETMAR ontology browser.

4 EIS Semantic Frameworks

In this chapter, we introduce a few projects that developed semantic frameworks or semantic framework architectures for environmental information systems (EIS) (c.f., section 4.1). The introduced semantic frameworks are then compared in terms of the capabilities and functionalities they support (c.f., section 4.2).

4.1 Existing EIS Semantic Frameworks

Many EIS semantic frameworks have been developed over the past few years for different purposes and with different capabilities and functionalities. This section introduces the main of these semantic frameworks and describes their features.

4.1.1 *OTEG Semantic Framework*

As part of the European Space Agency's OTEG (Open Access Ontology / Terminology for the GMES Space Component) project, Epistematica developed an ontology-based earth observation (EO) resources discovery framework[19]. The system is based on a set of interrelated ontologies that users can browse. The ontology browser shows terms definitions as well as relationships among terms. While browsing the ontologies, the system displays links to the datasets related to the terms being clicked.

The OTEG EO resources discovery framework also provides an interactive map of the ontology terms (c.f., Figure 4.1). Clicking a term allows to expand it, by displaying its sub-terms. Such an interactive map is very helpful for users as it provides a graphical and more intuitive view of the ontologies.

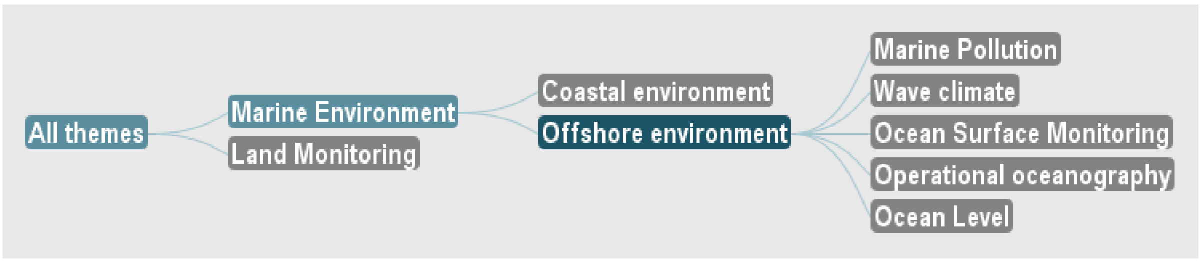


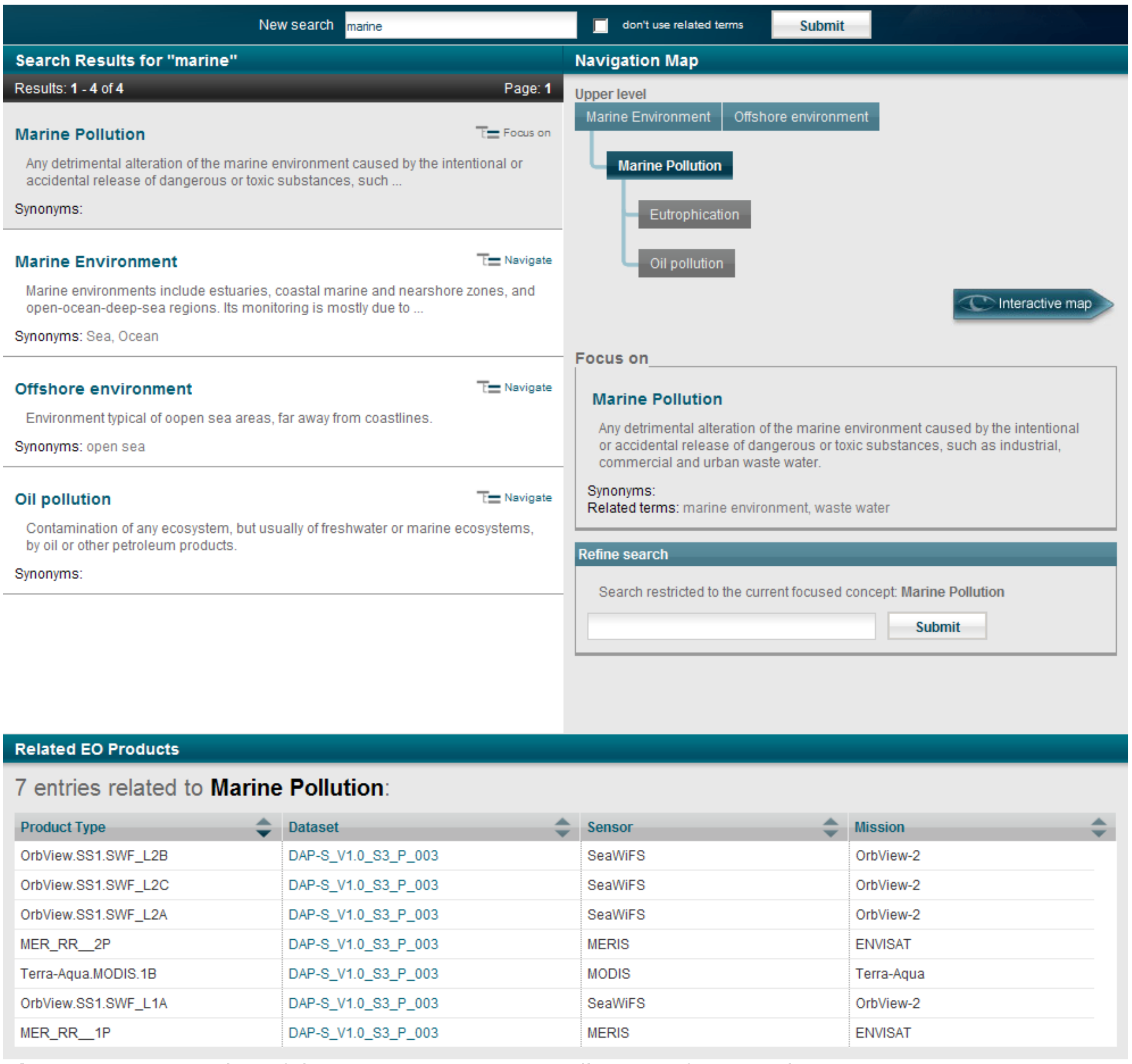
Figure 4.1. Screenshot of the OTEG interactive map

A screenshot of the OTEG EO resources discovery framework is shown in Figure 4.2. A search box allows free text search within the ontologies. Results of the search are listed in the top left division of the page. Results consist of ontology terms related to the text entered by the user, as well as definitions and synonyms. The top right division shows an interactive map of the ontology focused on the selected term from the list of results (here Marine Pollution).

The bottom division section of the page contains entries (EO products) related to the selected term (here Marine Pollution).

A "Refine Search" tool allows search restricted to the current focused concept (Marine Pollution).

[19] http://esaotewiki.epistematica.com/OTE/navigateInfoDomain



Search Results for "marine"

Results: **1 - 4** of **4** Page: **1**

Marine Pollution Focus on

Any detrimental alteration of the marine environment caused by the intentional or accidental release of dangerous or toxic substances, such ...

Synonyms:

Marine Environment Navigate

Marine environments include estuaries, coastal marine and nearshore zones, and open-ocean-deep-sea regions. Its monitoring is mostly due to ...

Synonyms: Sea, Ocean

Offshore environment Navigate

Environment typical of oopen sea areas, far away from coastlines.

Synonyms: open sea

Oil pollution Navigate

Contamination of any ecosystem, but usually of freshwater or marine ecosystems, by oil or other petroleum products.

Synonyms:

Navigation Map

Focus on

Marine Pollution

Any detrimental alteration of the marine environment caused by the intentional or accidental release of dangerous or toxic substances, such as industrial, commercial and urban waste water.

Synonyms:
Related terms: marine environment, waste water

Refine search

Search restricted to the current focused concept: **Marine Pollution**

Related EO Products

7 entries related to **Marine Pollution**:

Product Type	Dataset	Sensor	Mission
OrbView.SS1.SWF_L2B	DAP-S_V1.0_S3_P_003	SeaWiFS	OrbView-2
OrbView.SS1.SWF_L2C	DAP-S_V1.0_S3_P_003	SeaWiFS	OrbView-2
OrbView.SS1.SWF_L2A	DAP-S_V1.0_S3_P_003	SeaWiFS	OrbView-2
MER_RR__2P	DAP-S_V1.0_S3_P_003	MERIS	ENVISAT
Terra-Aqua.MODIS.1B	DAP-S_V1.0_S3_P_003	MODIS	Terra-Aqua
OrbView.SS1.SWF_L1A	DAP-S_V1.0_S3_P_003	SeaWiFS	OrbView-2
MER_RR__1P	DAP-S_V1.0_S3_P_003	MERIS	ENVISAT

Figure 4.2. Screenshot of the OTEG EO resources discovery framework

4.1.2 *InterRisk Semantic Framework*

Another semantic framework was developed under the EU FP6 InterRisk [20] project (September 2006 – August 2009) for the discovery of data and services (Web Feature Service (WFS), Web Map Service (WMS), Web Coverage Service (WCS) and CSW). The InterRisk semantic framework and ontologies build on the OTEG ones [CTL09]. The objective of the InterRisk semantic framework is to discover datasets and web services while searching and browsing a set of ontologies.

4.1.3 *ICAN Semantic Framework*

The International coastal Atlas Network, ICAN, is a network of scientists and organisations with interest in the coastal and marine domain. ICAN aims to be a global reference for the development of coastal web atlases.

[20] http://interrisk.nersc.no/

The ICAN community developed an ontology-based mediator for coastal web atlases. The current version of the ICAN prototype[21] supports the OGC Catalogue Service for the Web (CSW). Each atlas delivers metadata records through a CSW.

The ICAN approach uses ontologies for both facilitating interoperability and semantic data discovery. Metadata records for a given atlas use the terms of an ontology, called local ontology. For instance the Marine Irish Digital Atlas (MIDA) uses its own ontology; the Oregon Coastal Atlas (OCA) uses another one. The approach relies on a common ontology (global ontology) that defines common terms (keywords) for users of the ICAN prototype (c.f. Figure 4.3). Mappings between the global ontology and the local ones allow the mediator to translate terms from the global ontology to a local one.

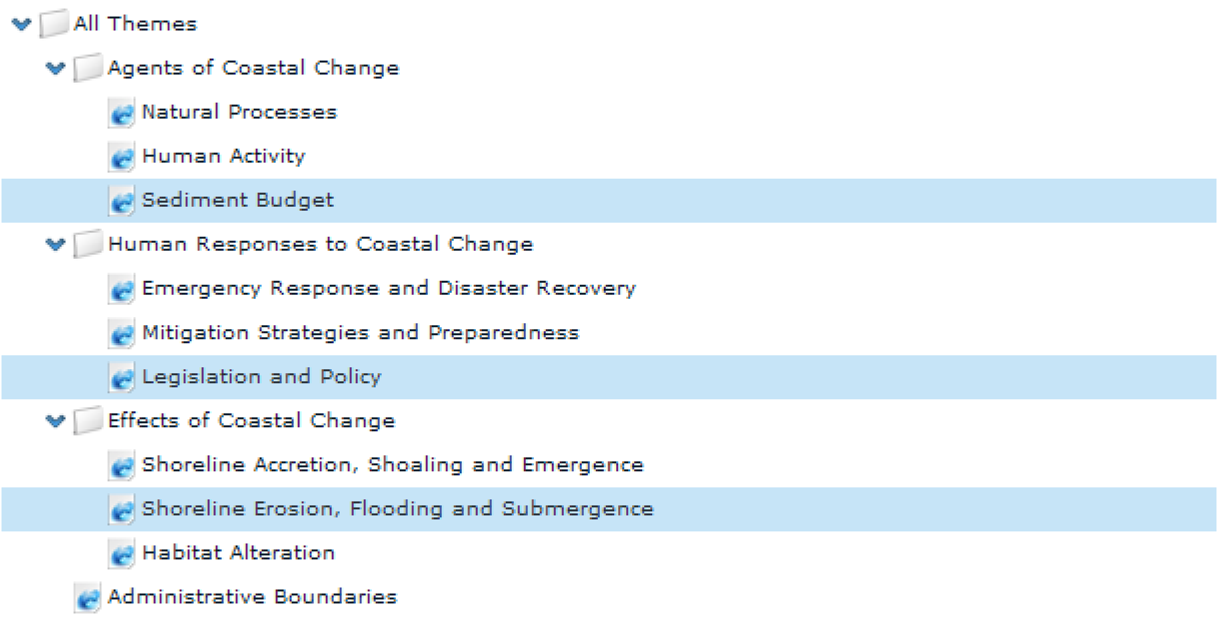
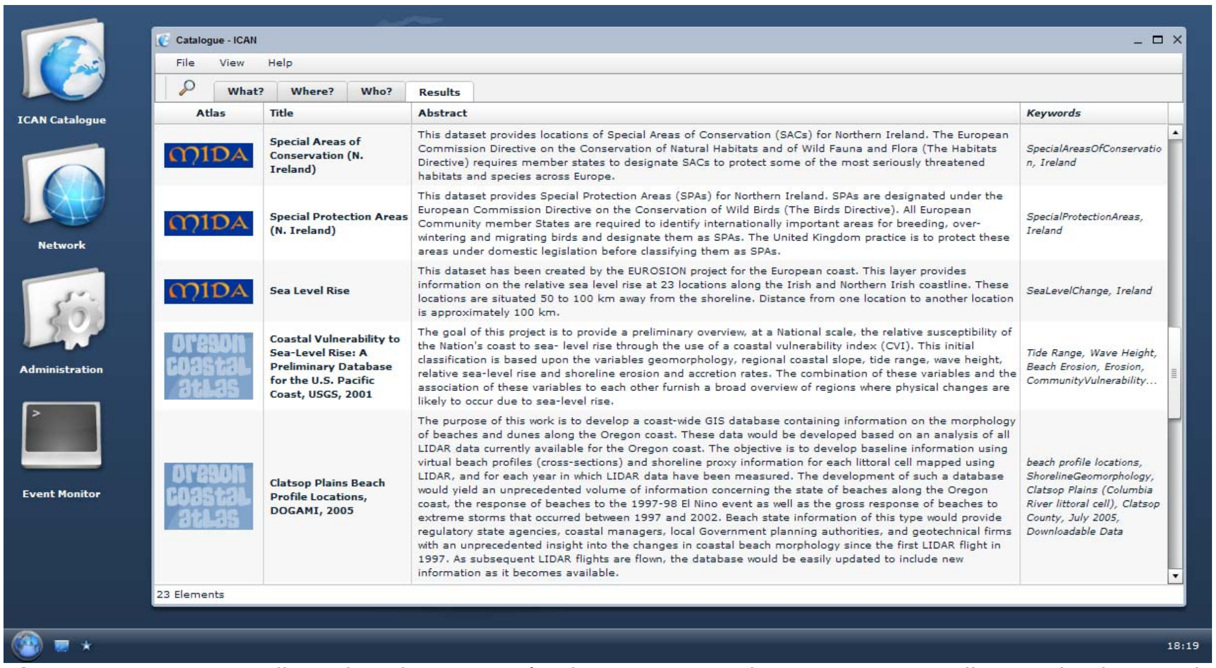

- All Themes
  - Agents of Coastal Change
    - Natural Processes
    - Human Activity
    - Sediment Budget
  - Human Responses to Coastal Change
    - Emergency Response and Disaster Recovery
    - Mitigation Strategies and Preparedness
    - Legislation and Policy
  - Effects of Coastal Change
    - Shoreline Accretion, Shoaling and Emergence
    - Shoreline Erosion, Flooding and Submergence
    - Habitat Alteration
  - Administrative Boundaries

Figure 4.3. ICAN global ontology: Terms highlighted in Blue are those selected by the user.

A user of the ICAN prototype may select one or more keywords and an area of interest (bounding box) and submit a search request to the mediator. The mediator translates the terms selected by the user into local terms and submits a request to the local atlases using terms from their ontologies. Results of such requests are then gathered and returned to the user (c.f. Figure 4.4).

[21] http://ican.ucc.ie



Atlas	Title	Abstract	Keywords
MIDA	Special Areas of Conservation (N. Ireland)	This dataset provides locations of Special Areas of Conservation (SACs) for Northern Ireland. The European Commission Directive on the Conservation of Natural Habitats and of Wild Fauna and Flora (The Habitats Directive) requires member states to designate SACs to protect some of the most seriously threatened habitats and species across Europe.	*SpecialAreasOfConservation, Ireland*
MIDA	Special Protection Areas (N. Ireland)	This dataset provides Special Protection Areas (SPAs) for Northern Ireland. SPAs are designated under the European Commission Directive on the Conservation of Wild Birds (The Birds Directive). All European Community member States are required to identify internationally important areas for breeding, over-wintering and migrating birds and designate them as SPAs. The United Kingdom practice is to protect these areas under domestic legislation before classifying them as SPAs.	*SpecialProtectionAreas, Ireland*
MIDA	Sea Level Rise	This dataset has been created by the EUROSION project for the European coast. This layer provides information on the relative sea level rise at 23 locations along the Irish and Northern Irish coastline. These locations are situated 50 to 100 km away from the shoreline. Distance from one location to another location is approximately 100 km.	*SeaLevelChange, Ireland*
Oregon Coastal Atlas	Coastal Vulnerability to Sea-Level Rise: A Preliminary Database for the U.S. Pacific Coast, USGS, 2001	The goal of this project is to provide a preliminary overview, at a National scale, the relative susceptibility of the Nation's coast to sea- level rise through the use of a coastal vulnerability index (CVI). This initial classification is based upon the variables geomorphology, regional coastal slope, tide range, wave height, relative sea-level rise and shoreline erosion and accretion rates. The combination of these variables and the association of these variables to each other furnish a broad overview of regions where physical changes are likely to occur due to sea-level rise.	*Tide Range, Wave Height, Beach Erosion, Erosion, CommunityVulnerability...*
Oregon Coastal Atlas	Clatsop Plains Beach Profile Locations, DOGAMI, 2005	The purpose of this work is to develop a coast-wide GIS database containing information on the morphology of beaches and dunes along the Oregon coast. These data would be developed based on an analysis of all LIDAR data currently available for the Oregon coast. The objective is to develop baseline information using virtual beach profiles (cross-sections) and shoreline proxy information for each littoral cell mapped using LIDAR, and for each year in which LIDAR data have been measured. The development of such a database would yield an unprecedented volume of information concerning the state of beaches along the Oregon coast, the response of beaches to the 1997-98 El Nino event as well as the gross response of beaches to extreme storms that occurred between 1997 and 2002. Beach state information of this type would provide regulatory state agencies, coastal managers, local Government planning authorities, and geotechnical firms with an unprecedented insight into the changes in coastal beach morphology since the first LIDAR flight in 1997. As subsequent LIDAR flights are flown, the database would be easily updated to include new information as it becomes available.	*beach profile locations, ShorelineGeomorphology, Clatsop Plains (Columbia River littoral cell), Clatsop County, July 2005, Downloadable Data*

23 Elements

Figure 4.4. ICAN mediator interface: Results from MIDA and OCA corresponding to the keywords selected in Figure 4.2.

Keywords translation in the ICAN mediator relies on an inference engine that ensures that narrower terms are considered.

4.1.4 *MMI Semantic Framework*

The Marine Metadata Interoperability (MMI)[22] project aims to promote collaborative research in the marine science domain, by simplifying the metadata into specific, straightforward guidance. MMI developed a semantic framework for facilitating data interoperability in the marine science community. The MMI Semantic Framework[23] consists of a set of tools that allow users to work with semantic technologies, and a set of guidance documents, worked with the marine science community to establish a set of best practices.

As shown in Figure 4.5, the OOI CI Semantic Framework allows data and semantics providers to create and register vocabularies and to register mappings between vocabularies. Data users may request vocabularies and issue semantic queries.

[22] http://marinemetadata.org

[23] http://marinemetadata.org/semanticframework

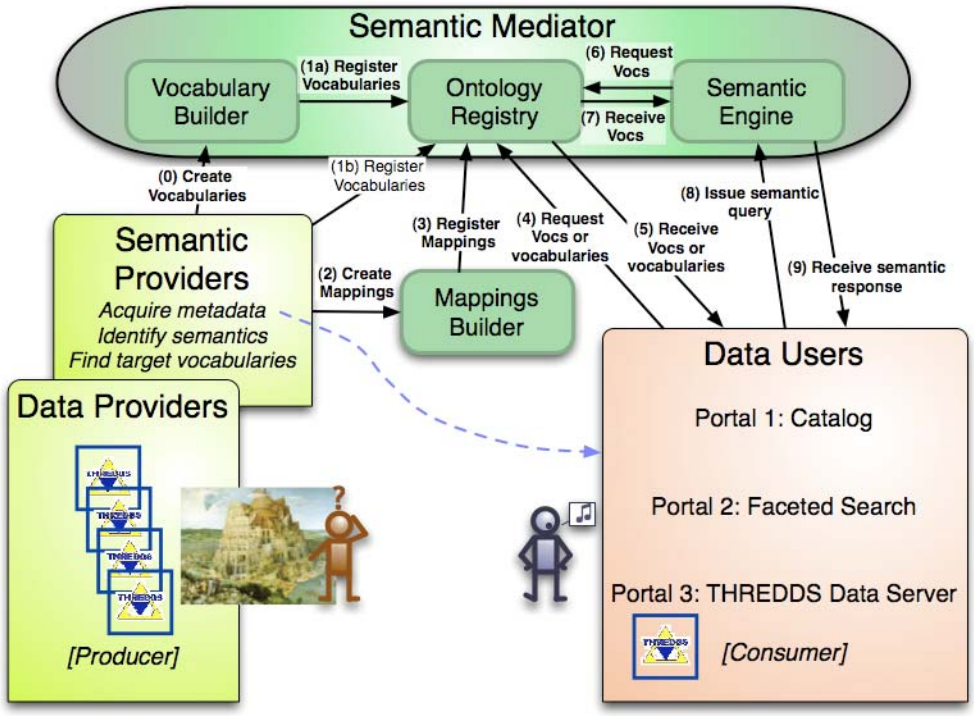


Figure 4.5. MMI Semantic Framework Architecture (Source: Architectural Framework and Operational Concept for Semantic Interoperability: http://marinemetadata.org/semanticframeworkconcept)

Several other functionalities are available for data providers. For instance, data providers can feed the Semantic Framework with content by providing ontologies and related data. Examples of functionalities available for data providers are:

- Convert an existing controlled vocabulary into an ontology and register it with the MMI Ontology Registry & Repository;
- Provide mappings between local ontologies and community-supported ontologies;
- Make data available with associated metadata stored in a registered ontology.

4.1.5 *OOI CI Semantic Framework*

The Ocean Observatories Initiative (OOI)[24] is a US project aiming to construct a networked infrastructure of science-driven sensor systems to measure the physical, chemical, geological and biological variables in the ocean and seafloor. The OOI Cyberinfrastructure (OOI CI)[25] constitutes the integrating element that links and binds the physical infrastructure into a coherent system-of-systems. As part of the OOI CI Data Management subsystem, a semantic framework is being developed. The OOI CI Semantic Framework[26] [Be09] aims to demonstrate a number of end-to-end operations and capabilities that can semantically enable the OOI. The OOI CI semantic framework is based on the MMI semantic framework (c.f., subsection 4.1.4).

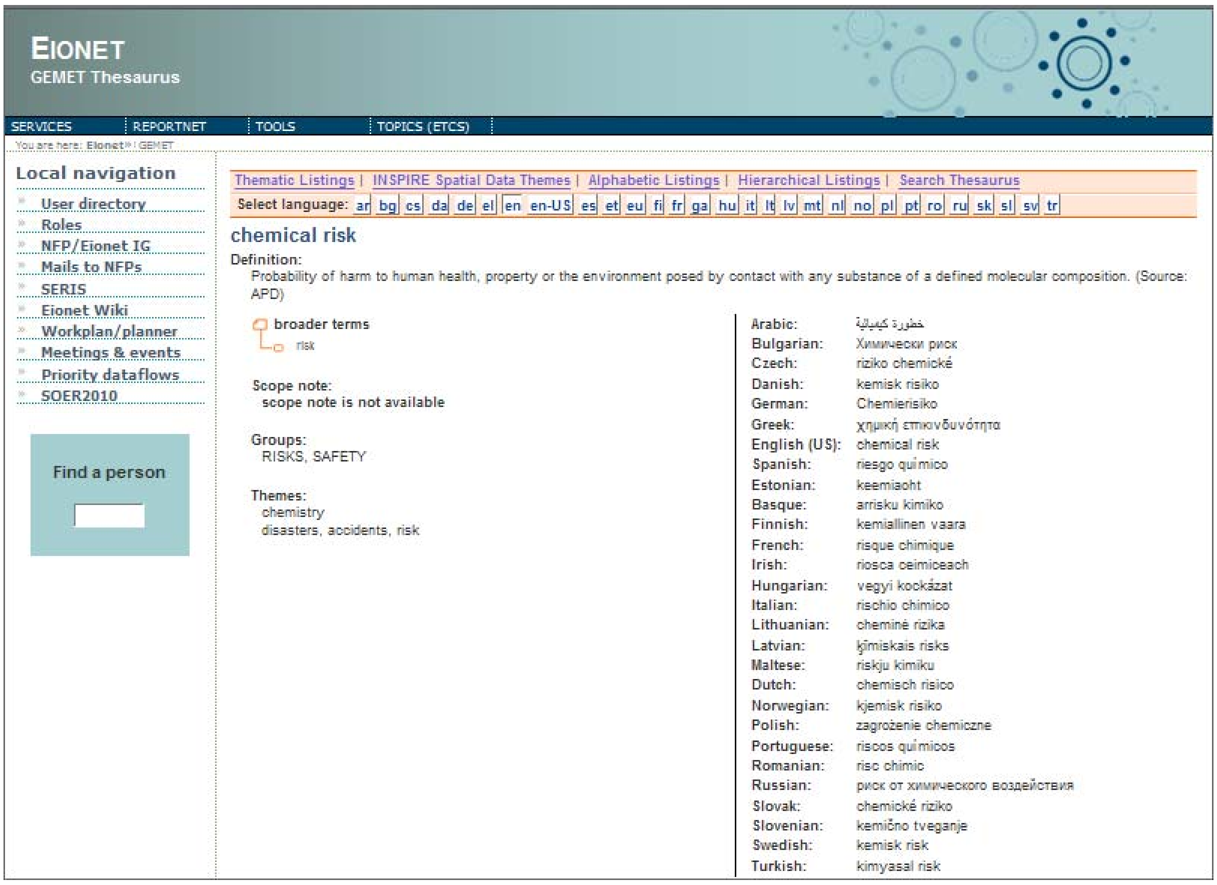
[24] http://www.oceanleadership.org/programs-and-partnerships/ocean-observing/ooi/

[25] http://ci.oceanobservatories.org/

[26] http://oceanobservatories.org/spaces/display/CIDev/Semantic+Framework+Integration

4.1.6 *GEMET*

The General Multilingual Environmental Thesaurus (GEMET)[27] has been developed as an indexing, retrieval and control tool for the European Topic Centre on Catalogue of Data Sources (ETC/CDS) and the European Environment Agency (EEA), Copenhagen. GEMET is multilingual, with support for 29 languages. The GEMET graphical user interface (c.f., Figure 4.6) provides users with various vocabulary listing options (thematic, alphabetic and hierarchical). It supports free text search with the possibility of selecting the language of interest. Definition, related terms (such as narrower and broader terms) and translations are displayed when a term is selected.


EIONET
GEMET Thesaurus
SERVICES
REPORTNET
TOOLS
TOPICS (ETCS)
You are here: Eionet » GEMET
Local navigation
User directory
Roles
NFP/Eionet IG
Mails to NFPs
SERIS
Eionet Wiki
Workplan/planner
Meetings & events
Priority dataflows
SOER2010
Find a person
Thematic Listings | INSPIRE Spatial Data Themes | Alphabetic Listings | Hierarchical Listings | Search Thesaurus
Select language: ar bg cs da de el en en-US es et eu fi fr ga hu it lt lv mt nl no pl pt ro ru sk sl sv tr

chemical risk

Definition:
Probability of harm to human health, property or the environment posed by contact with any substance of a defined molecular composition. (Source: APD)

broader terms
risk

Scope note:
scope note is not available

Groups:
RISKS, SAFETY

Themes:
chemistry
disasters, accidents, risk

Arabic:	خطورة كيميائية
Bulgarian:	Химически риск
Czech:	riziko chemické
Danish:	kemisk risiko
German:	Chemierisiko
Greek:	χημική επικινδυνότητα
English (US):	chemical risk
Spanish:	riesgo químico
Estonian:	keemiaoht
Basque:	arrisku kimiko
Finnish:	kemiallinen vaara
French:	risque chimique
Irish:	riosca ceimiceach
Hungarian:	vegyi kockázat
Italian:	rischio chimico
Lithuanian:	cheminė rizika
Latvian:	ķīmiskais risks
Maltese:	riskju kimiku
Dutch:	chemisch risico
Norwegian:	kjemisk risiko
Polish:	zagrożenie chemiczne
Portuguese:	riscos químicos
Romanian:	risc chimic
Russian:	риск от химического воздействия
Slovak:	chemické riziko
Slovenian:	kemično tveganje
Swedish:	kemisk risk
Turkish:	kimyasal risk

Figure 4.6. The GEMET Graphical User Interface showing a term (chemical risk), its definition, translations and related terms

The backbone of GEMET is a SKOS thesaurus expressed in an RDF structure. The GEMET data are exposed to remote applications in the RDF/XML format via web services. Currently, the API for the GEMET web services is undergoing changes and its specification[28] is available as a draft. The current version of the API supports a number of methods such as `GetTopmostConcepts` for retrieving the top concepts of a thesaurus, `getRelatedConcepts` for retrieving the list of concepts with a given relation to a given concept, or `getAllTranslationsForConcept` for retrieving all translations for a property of a given concept.

[27] http://www.eionet.europa.eu/gemet

[28] https://svn.eionet.europa.eu/projects/Zope/wiki/GEMETWebServiceAPI

4.1.7 *ENVISION and SWING*

The objective of the Semantic Web services Interoperability for Geospatial decision making (SWING) project (2006 to 2009) was to deploy the semantic web service (SWS) technology in the geospatial domain. It particularly focused on reducing the complexity of creating semantic descriptions and increasing the number of semantically described services.

The SWING project developed a set of tools, among which are:

- Visual OntoBridge, which provides a graphical user interface for annotating WFS schemas and querying ontology concepts and triples;
- Concept Repository which is a service that provides access to the domain ontologies developed for the SWING project;
- Web Service Execution Environment (WSMX)[29] which is an execution environment designed to perform discovery, mediation, invocation and interoperation of semantic web services; WSMX is the reference implementation of the Web Service Modelling Ontology (WSMO)[30], an ontology for describing various aspects related to Semantic Web service.

The follow-up of SWING is the ENVISION project[31] which has recently started. ENVISION provides an ENVIronmental Services Infrastructure with ONtologies that aims to support non ICT-skilled users in the process of semantic discovery and adaptive chaining and composition of environmental services.

4.1.8 *ORCHESTRA Semantic Catalogue Architecture*

The Open Architecture and Spatial Data Infrastructure for Risk Management project (ORCHESTRA)[32] aims at designing and implementing an open, service-oriented software architecture to overcome the interoperability problems in the domain of multi-risk management. A reference model for the ORCHESTRA architecture [Us07, KS09] has been produced and accepted as an OGC best practice. It proposes an architecture for a Semantic Catalogue the aim of which is to improve the search for resources by exploiting the semantic relationships between concepts defined in an ontology. The proposed architecture is illustrated in Figure 4.7.

[29] http://www.wsmx.org/

[30] http://www.wsmo.org/

[31] http://www.envision-project.eu/

[32] http://www.eu-orchestra.org/overview.shtml

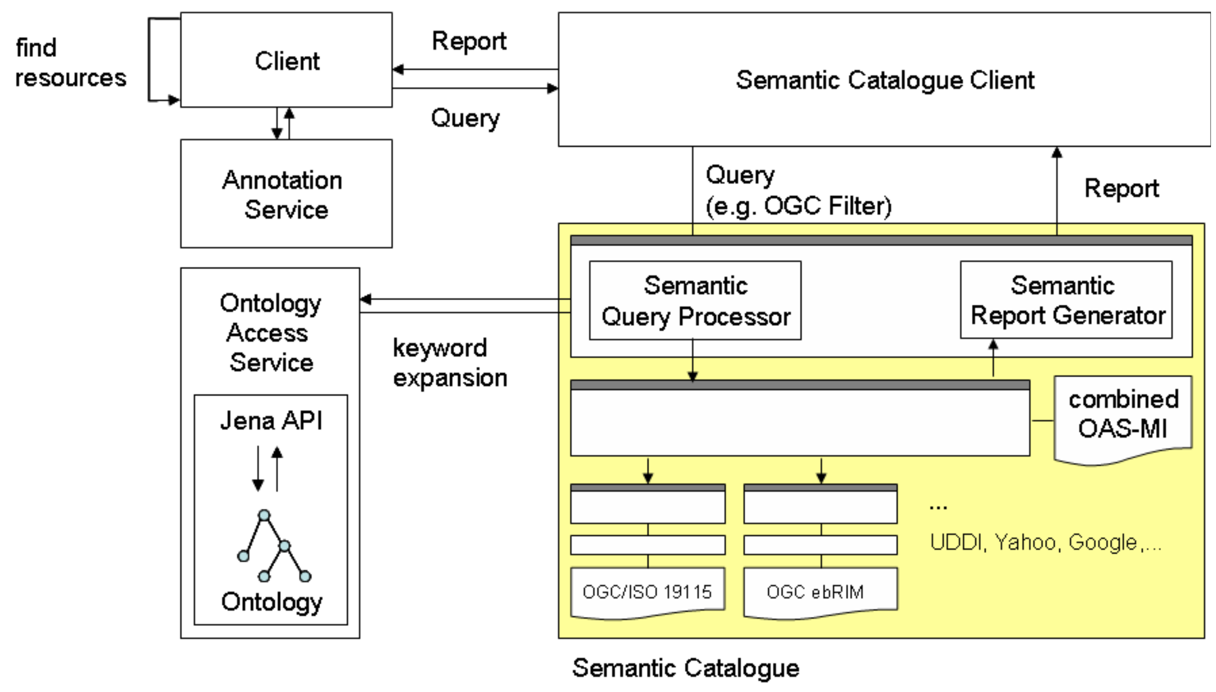


Figure 4.7. The ORCHESTRA Semantic Catalogue Architecture (source: [Us07])

The proposed Semantic Catalogue is very similar to the ICAN prototype. It suggests that (a subset of) the thematic domain of the user be represented in the ontology. On the front-end to a client application, the Semantic Catalogue provides an interface in form of the ORCHESTRA Catalogue Service. On the back-end, it offers access to various, heterogeneous catalogue services. Access to these services is transparent to the user of the Semantic Catalogue. Like in the ICAN prototype, a user query is first analysed in a semantic query processor that uses the Ontology Access Service to expand the query according to related concepts in the ontology. The user query is then rewritten according to each individual catalogue service and executed. Responses are then grouped assembled and structured by a semantic report generator and returned as a query response to the client.

The ORCHESTRA architecture further proposes an Annotation Service for annotating selected textual results against the ontology that has been used in the query expansion in order to assess and interpret the results in the context of the thematic domain.

4.1.9 *OOSTethys*

OOSTethys[33] is a provider-to-user data systems framework for enabling data discovery and access. The framework uses a network of interoperable standards to deliver ocean observations and to establish robust data exchange.

The OOSTethys system architecture, illustrated in Figure 4.8, consists of the following components.

1. Data Provider for serving observation data,
2. Semantic Mediator for registering vocabularies and mapping between vocabularies and for performing updates and queries on existing vocabularies,
3. Service Registry for allowing service registration and discovery,
4. Data Aggregator for temporary storage of data from Data Provider, which will be used in creating post products,

[33] http://www.oostethys.org/

5. Data Archiver, a data storage site that acquires data from Data Providers and Data Aggregators, useful for establishing a policy for data retention, and a manner to ensure data provenance
6. Visualisation Portal for the end-user interaction with the OOSTethys system.

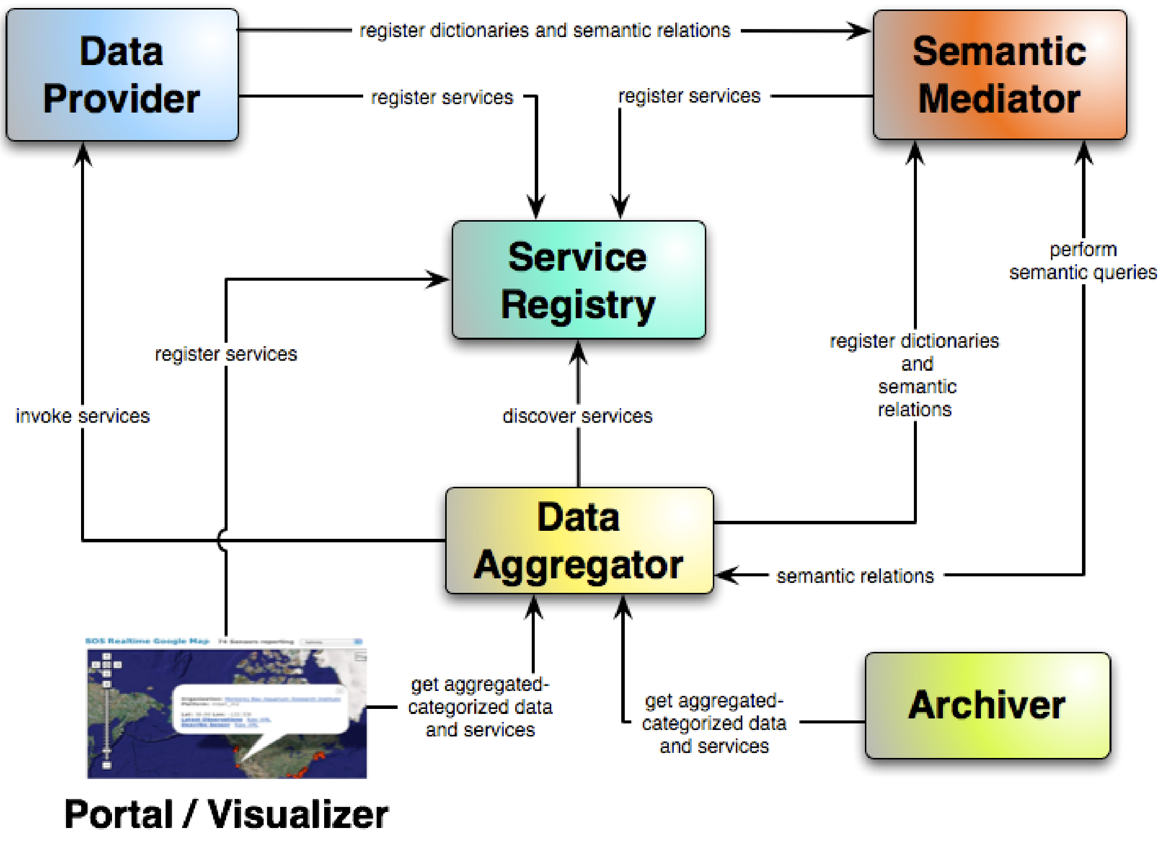


Figure 4.8. The OOSTethys System Architecture (source: OOSTethys Architecture Web Page: http://www.oostethys.org/System%20Architecture)

4.1.10 *Virtual Solar Terrestrial Observatory*

The Virtual Solar Terrestrial Observatory (VSTO)[34] is a unified semantic environment serving data from diverse data archives in the fields of solar, solar-terrestrial, and space physics (SSTSP).

As shown in Figure 4.9, VSTO currently serves data from two data archives: CEDAR[35] (Coupling, Energetics and Dynamics of Atmospheric Regions) and MLSO[36] (Mauna Loa Solar Observatory).

[34] http://www.vsto.org/

[35] http://www.nsf.gov/pubs/2005/nsf05554/nsf05554.htm

[36] http://mlso.hao.ucar.edu/cgi-bin/mlso_homepage.cgi

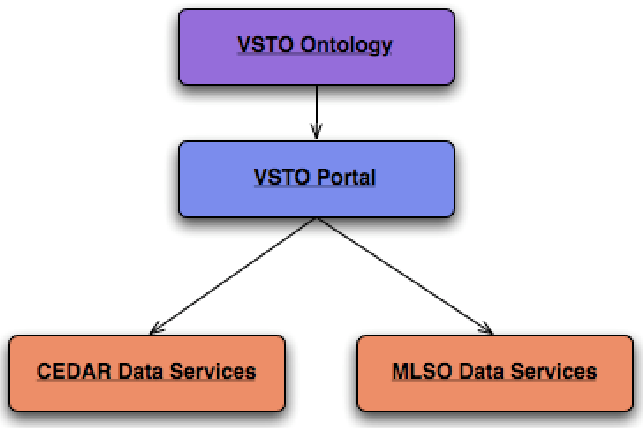


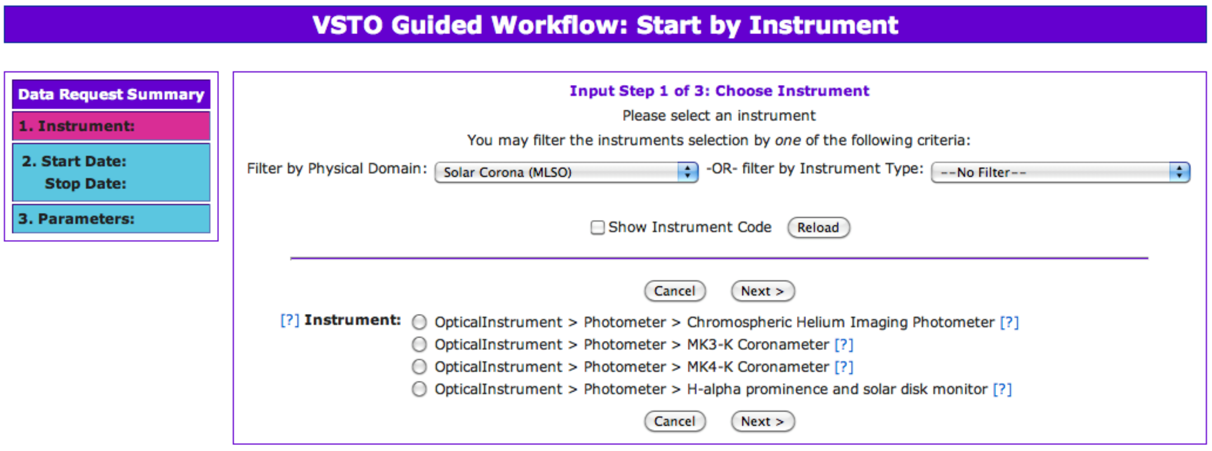
Figure 4.9. The VSTO portal currently allows discovery and access of data from the CEDAR and MLSO data archives. Discovery relies on the VSTO family of ontologies. (source: VSTO Web Page: http://www.vsto.org)

Data discovery in VSTO relies on a family of ontologies representing the SSTSP domains knowledge. The VSTO ontologies include physical domains, instruments and physical parameters. The VSTO family of ontologies consists of:

- A container ontology that includes all the following ontologies
- An ontology containing the VSTO core classes and relationships (VSTO ontology structure)
- CEDAR specific instances
- MLSO specific instances

All these ontologies are available online from the VSTO website.

Figure 4.10 shows a screenshot of the VSTO portal where a list of instruments is displayed. The list of instruments may be filtered by physical domain (e.g. Solar Corona) of by instrument type. The "[?]" link following each term is meant to display the definition for the corresponding term. Although not all terms have definitions (most do not), the functionality is there. By selecting on instrument and clicking the "Next" button, the user is asked to select the time period of interest, then is provided with the relevant list of products which may be accessed in different formats.


VSTO Guided Workflow: Start by Instrument

Data Request Summary
1. Instrument:
2. Start Date:
Stop Date:
3. Parameters:

Input Step 1 of 3: Choose Instrument
Please select an instrument
You may filter the instruments selection by *one* of the following criteria:
Filter by Physical Domain: Solar Corona (MLSO) -OR- filter by Instrument Type: --No Filter--

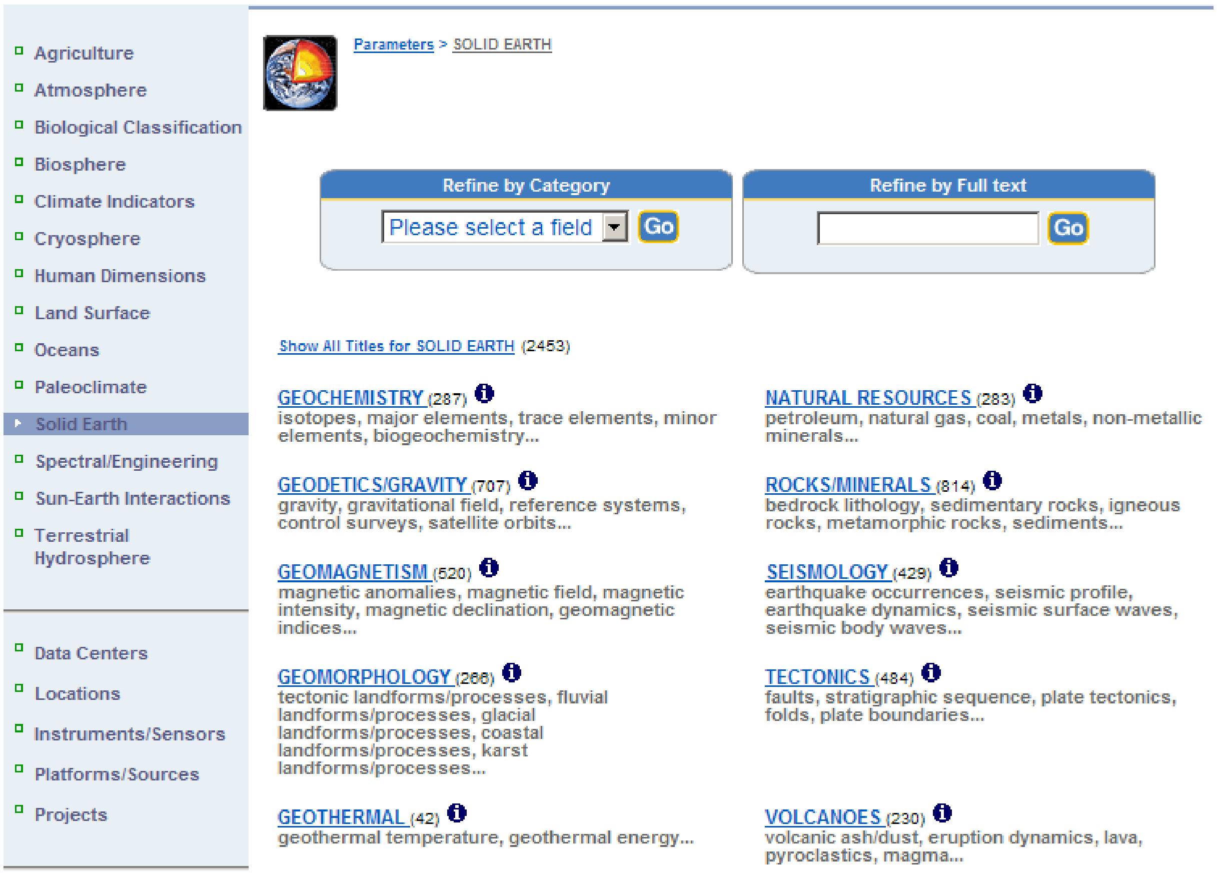
Show Instrument Code

[?] Instrument:
- OpticalInstrument > Photometer > Chromospheric Helium Imaging Photometer [?]
- OpticalInstrument > Photometer > MK3-K Coronameter [?]
- OpticalInstrument > Photometer > MK4-K Coronameter [?]
- OpticalInstrument > Photometer > H-alpha prominence and solar disk monitor [?]

Figure 4.9. Screenshot of The VSTO portal showing a list of instruments filtered by physical domain (Solar Corona)

4.1.11 *NASA Global Change Master Directory (GCMD)*

NASA Global Change Master Directory (GCMD)[37] is an entire data portal as well as a set of descriptors and broader content metadata files using the Directory Interchange Format (DIF)[38]. DIF is a descriptive and standardised format for exchanging information about scientific data sets, conceived in 1987.
GCMD offers a semantic framework for discovering data using a thesaurus of keywords related to themes, data centres, locations, instruments, platforms and projects. Figure 4.10 shows a screenshot of GCMD interface. The top left division contains the list of categories, "Solid Earth" being the one selected. The bottom left one contains the list of keyword types (data centres, locations, etc.). The right division shows the terms belonging to the selected category. The number of results matching a term is given as an indication. Users can view the definition of a term by clicking the information symbol (i) or simply view the sub terms by clicking the term. In this way, users can navigate from a given category through a hierarchy of terms down to results that correspond to matching datasets.



Show All Titles for SOLID EARTH (2453)

GEOCHEMISTRY (287)
isotopes, major elements, trace elements, minor elements, biogeochemistry...

NATURAL RESOURCES (283)
petroleum, natural gas, coal, metals, non-metallic minerals...

GEODETICS/GRAVITY (707)
gravity, gravitational field, reference systems, control surveys, satellite orbits...

ROCKS/MINERALS (814)
bedrock lithology, sedimentary rocks, igneous rocks, metamorphic rocks, sediments...

GEOMAGNETISM (520)
magnetic anomalies, magnetic field, magnetic intensity, magnetic declination, geomagnetic indices...

SEISMOLOGY (429)
earthquake occurrences, seismic profile, earthquake dynamics, seismic surface waves, seismic body waves...

GEOMORPHOLOGY (266)
tectonic landforms/processes, fluvial landforms/processes, glacial landforms/processes, coastal landforms/processes, karst landforms/processes...

TECTONICS (484)
faults, stratigraphic sequence, plate tectonics, folds, plate boundaries...

GEOTHERMAL (42)
geothermal temperature, geothermal energy...

VOLCANOES (230)
volcanic ash/dust, eruption dynamics, lava, pyroclastics, magma...

Figure 4.10 A Screenshot of the NASA GCMD User Interface

4.1.12 *USGS Enterprise Web Document Catalog (EWDC)*

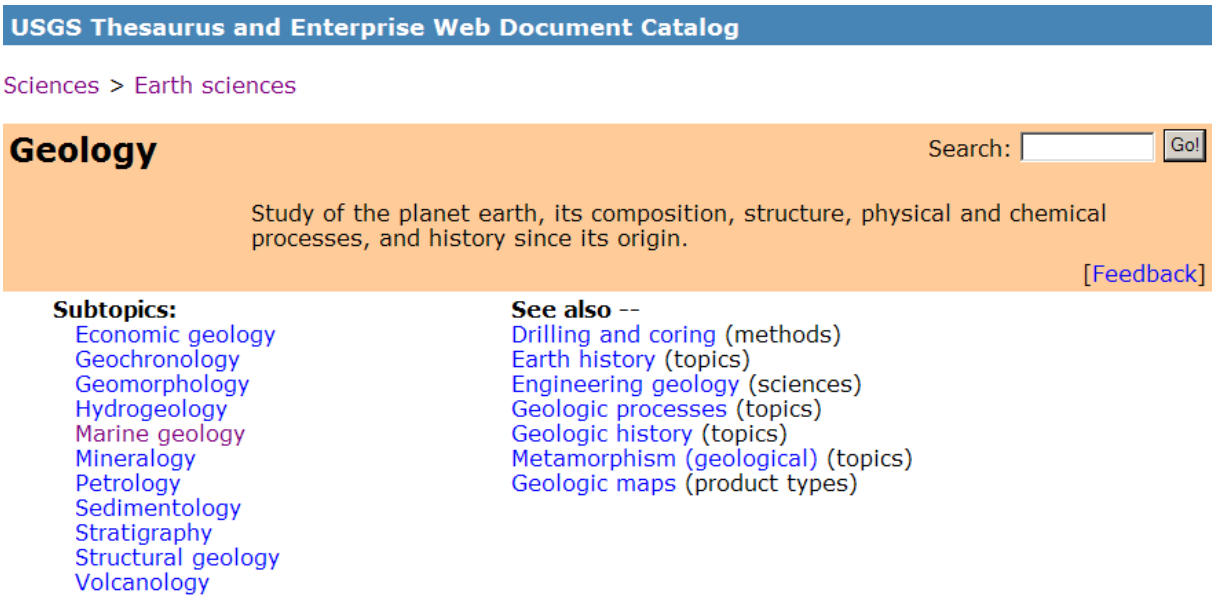
The USGS Thesaurus and Enterprise Web Document Catalog[39] provides links to the operational aspects of the public document catalogue for the USGS Enterprise Web. The catalogue makes full use of the USGS Thesaurus and several additional controlled vocabularies. This site is provided to permit interested outsiders to learn about the USGS use of these controlled vocabularies for categorising and finding information resources.

[37] http://gcmd.nasa.gov/

[38] http://gcmd.nasa.gov/User/difguide/difman.html

[39] http://geo-nsdi.er.usgs.gov/thesaurus/catalog/term.php

The USGS Thesaurus and Enterprise Web Document Catalog allows users to search and browse the USGGS thesaurus, view definitions of terms as well related terms (c.f., Figure 4.11).


USGS Thesaurus and Enterprise Web Document Catalog

Sciences > Earth sciences

Geology

Search: Go!

Study of the planet earth, its composition, structure, physical and chemical processes, and history since its origin.

[Feedback]

Subtopics:
Economic geology
Geochronology
Geomorphology
Hydrogeology
Marine geology
Mineralogy
Petrology
Sedimentology
Stratigraphy
Structural geology
Volcanology

See also --
Drilling and coring (methods)
Earth history (topics)
Engineering geology (sciences)
Geologic processes (topics)
Geologic history (topics)
Metamorphism (geological) (topics)
Geologic maps (product types)

Figure 4.11. Screenshot of the USGS Thesaurus and Enterprise Web Document Catalog

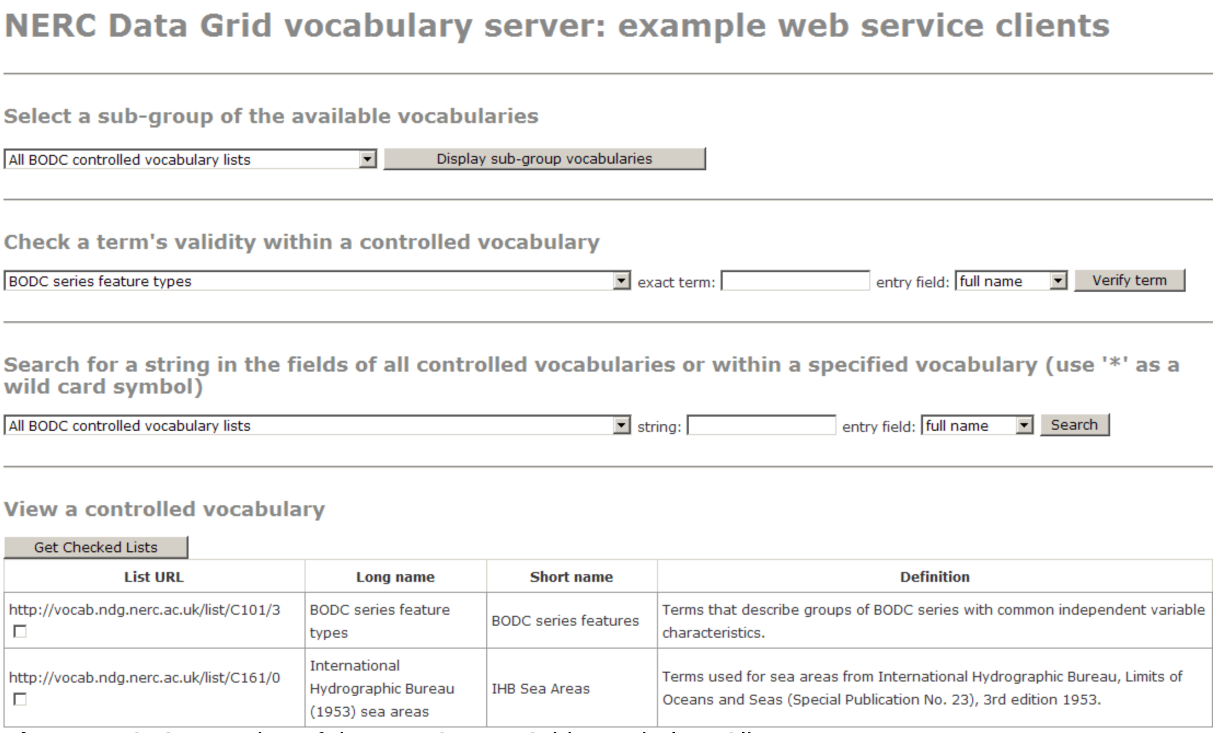
4.1.13 *NERC and SeaDataNet Vocabulary Servers*

The Natural Environment Research Council (NERC) has a vocabulary server for providing access to lists of standardised terms covering a broad spectrum of disciplines of relevance to the oceanographic and wider community. The NERC Data Grid Vocabulary Server[40] (NVS) provides the following methods:

- **`getList`** which retrieves a specific term and/or all terms on a given list,
- **`getMap`** which retrieves all terms from one or more lists, including their mappings to either one or more specified vocabularies or all vocabularies in the server,
- **`getRelatedRecordByCriteria`** for retrieving subject terms matching the given criteria and object terms matching the given predicate,
- **`getRelatedRecordByTerm`** which retrieves the subject term(s) referred to by the `subjectTerm` parameter and the object terms matching the given predicate,
- **`searchVocab`** which retrieves records containing a specified string in the term, abbreviated term or definition,
- **`verifyTerm`** for verifying that a term, an abbreviated term or a term definition exists in a given list,
- **`whatLists`** for retrieving information (URL, name, abbreviation, definition, version number and version date) on all visible lists or all visible lists in a particular category,
- **`whatListsCat`** which retrieves the terms that describe the vocabulary server subject categories.

Figure 4.12 shows a screenshot of a NERC vocabulary client which uses some of the above-listed methods.

[40] http://www.bodc.ac.uk/products/web_services/vocab/


NERC Data Grid vocabulary server: example web service clients

Select a sub-group of the available vocabularies

All BODC controlled vocabulary lists | Display sub-group vocabularies

Check a term's validity within a controlled vocabulary

BODC series feature types | exact term: | entry field: full name | Verify term

Search for a string in the fields of all controlled vocabularies or within a specified vocabulary (use '*' as a wild card symbol)

All BODC controlled vocabulary lists | string: | entry field: full name | Search

View a controlled vocabulary

Get Checked Lists

List URL	Long name	Short name	Definition
http://vocab.ndg.nerc.ac.uk/list/C101/3 ☐	BODC series feature types	BODC series features	Terms that describe groups of BODC series with common independent variable characteristics.
http://vocab.ndg.nerc.ac.uk/list/C161/0 ☐	International Hydrographic Bureau (1953) sea areas	IHB Sea Areas	Terms used for sea areas from International Hydrographic Bureau, Limits of Oceans and Seas (Special Publication No. 23), 3rd edition 1953.

Figure 4.12. Screenshot of the NERC Data Grid Vocabulary Client

The NERC Data Grid Vocabulary Server is also the platform used by the SeaDataNet[41] project for delivering their vocabularies. SeaDataNet is supported by the European Commission within the 6th Framework Programme (FP6), Integrated Research Infrastructure Initiative (I3), 2006 to 2011, to provide interoperable data across Europe and the Mediterranean.

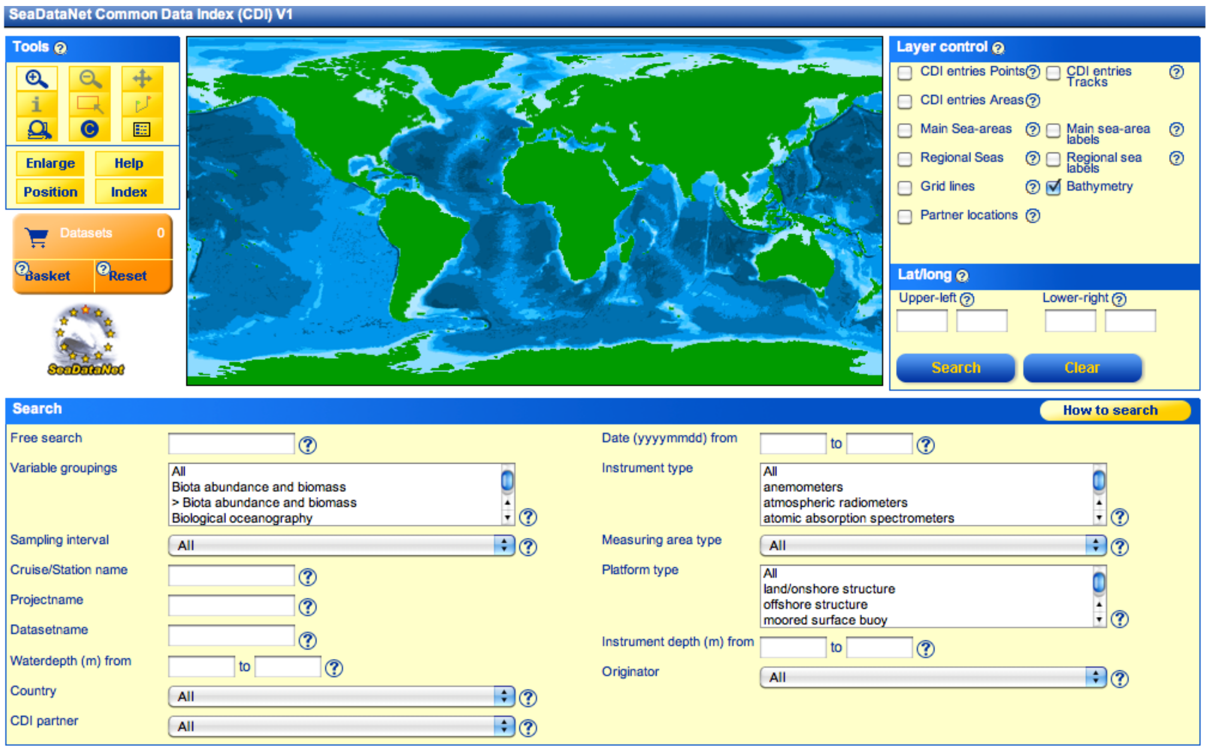
A data discovery client based on the NERC Vocabulary Server is available at http://seadatanet.maris2.nl/v_cdi_v1/search.asp (c.f. Figure 4.13). Other lists in the NVS support the 'Instrument type' and 'Platform type' facets of the discovery search.

The NVS contains a number of mappings between parameter vocabularies, for example between the Climate and Forecast (CF) Standard Names[42] and the SeaDataNet Parameter Discovery. Such vocabulary mappings provide the basis for semantic interoperability of vocabularies.

The NVS contains mappings between parameter vocabularies and units of measure. It is a short content development step from units of measure to canonical dimensionality (length, volume, temperature, etc.), which would provide one basis of semantic validation of the linkages between service chains through concept addressing as URLs, which NVS already supports.

[41] http://www.seadatanet.org/

[42] http://cf-pcmdi.llnl.gov/documents/cf-standard-names/



Layer control
- ☐ CDI entries Points ☐ CDI entries Tracks
- ☐ CDI entries Areas
- ☐ Main Sea-areas ☐ Main sea-area labels
- ☐ Regional Seas ☐ Regional sea labels
- ☐ Grid lines ☑ Bathymetry
- ☐ Partner locations

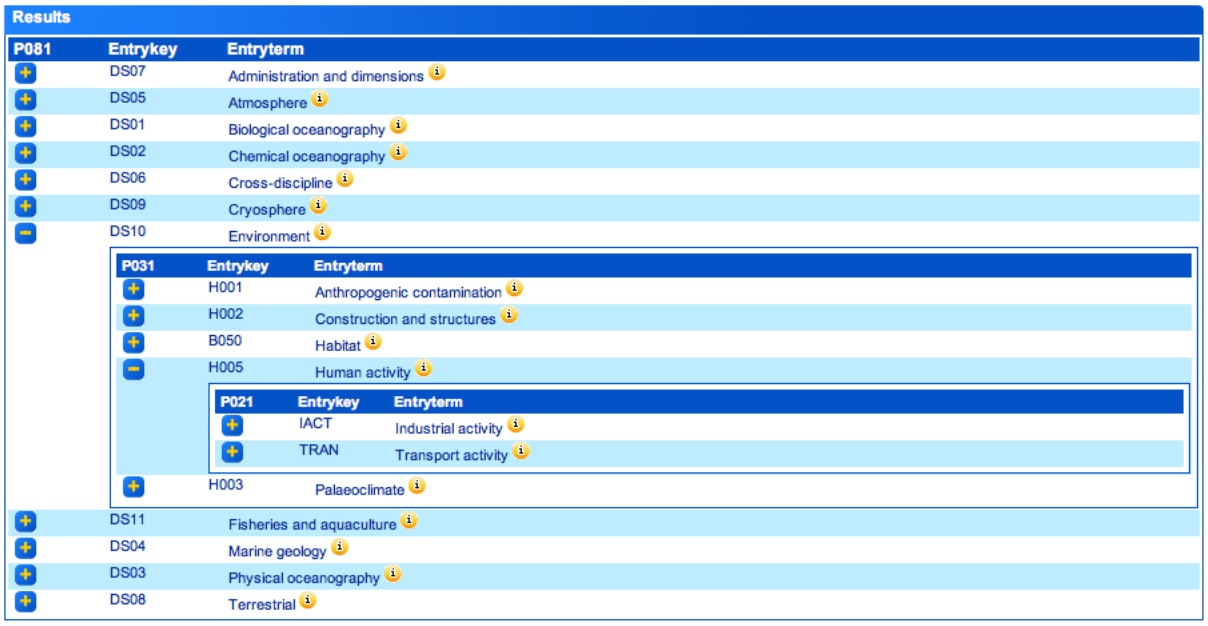
Lat/long
Upper-left | Lower-right

Search

Free search		Date (yyyymmdd) from	to
Variable groupings	All / Biota abundance and biomass / > Biota abundance and biomass / Biological oceanography	Instrument type	All / anemometers / atmospheric radiometers / atomic absorption spectrometers
Sampling interval	All	Measuring area type	All
Cruise/Station name		Platform type	All / land/onshore structure / offshore structure / moored surface buoy
Projectname			
Datasetname		Instrument depth (m) from	to
Waterdepth (m) from	to	Originator	All
Country	All		
CDI partner	All		

Figure 4.13. Screenshot of the SeaDataNet Data Discovery Client, based on the NVS

The NVS API provides support for ontology browsing clients. Currently two are known: the Maris thesaurus search[43], and various tools at http://vocab.ndg.nerc.ac.uk/. Figure 4.14 shows a screenshot of the Maris thesaurus search. Terms are organised in a hierarchy where nodes can be expanded or collapsed. Definitions of terms, abbreviations and dates of modification can be viewed by clicking the yellow "i" buttons next to the terms.


Results

P081	Entrykey	Entryterm
+	DS07	Administration and dimensions
+	DS05	Atmosphere
+	DS01	Biological oceanography
+	DS02	Chemical oceanography
+	DS06	Cross-discipline
+	DS09	Cryosphere
−	DS10	Environment

P031	Entrykey	Entryterm
+	H001	Anthropogenic contamination
+	H002	Construction and structures
+	B050	Habitat
−	H005	Human activity

P021	Entrykey	Entryterm
+	IACT	Industrial activity
+	TRAN	Transport activity

P031	Entrykey	Entryterm
+	H003	Palaeoclimate

P081	Entrykey	Entryterm
+	DS11	Fisheries and aquaculture
+	DS04	Marine geology
+	DS03	Physical oceanography
+	DS08	Terrestrial

Figure 4.14. Screenshot of the Maris thesaurus search, based on the NVS

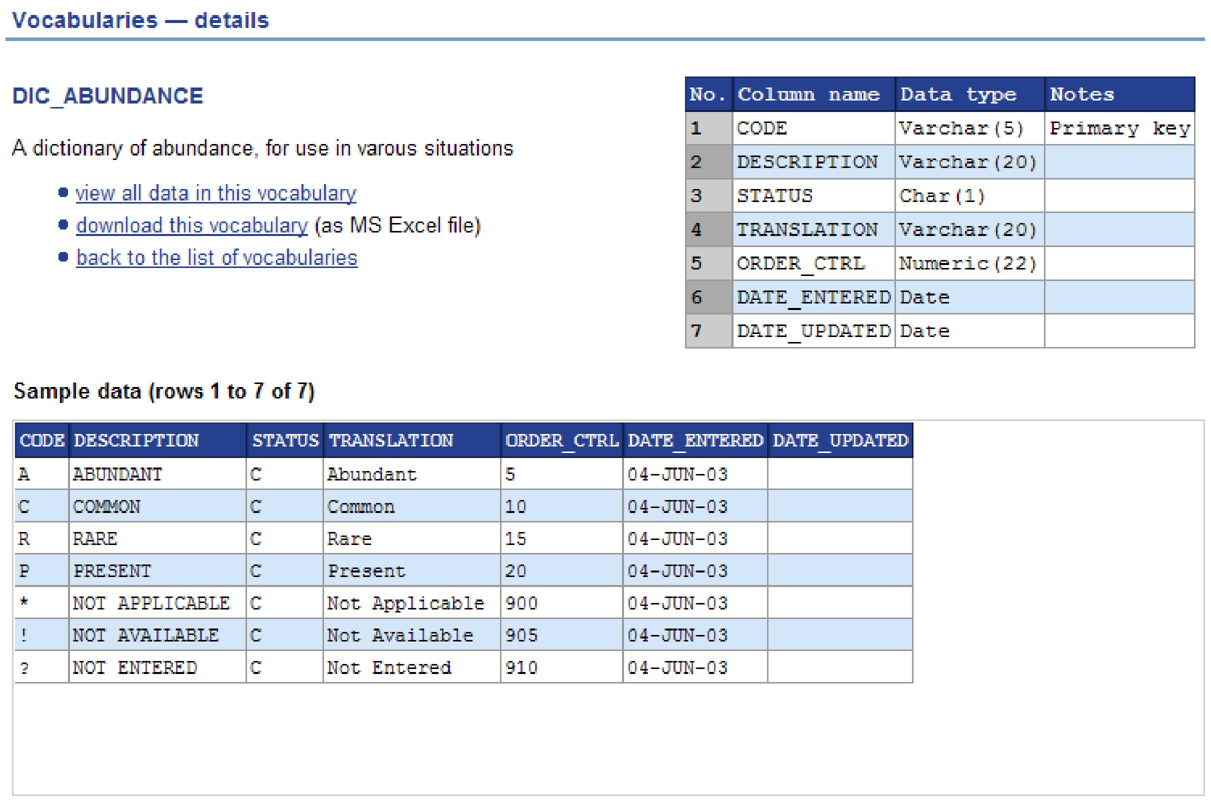
[43] http://seadatanet.maris2.nl/v_bodc_vocab/vocabrelations.aspx

Smart Search is specifically supported by the `getRelatedRecordByCriteria` method of the API.

Ontology registration is deliberately a manual process with the NVS to ensure that users conform to the high governance standards required by NVS.

4.1.14 *BGS Vocabulary Web Service*

The British Geological Survey (BGS) has a wide range of controlled vocabularies (currently[44] 211 vocabularies) available through the BGS Vocabulary Web Service[45]. The BGS Vocabulary Web Service enables users to discover and download the BGS vocabularies. It provides descriptions for the vocabulary lists. Figure 4.15 shows a screenshot of the BGS Vocabulary Web Service.



Vocabularies — details

DIC_ABUNDANCE

A dictionary of abundance, for use in varous situations

- view all data in this vocabulary
- download this vocabulary (as MS Excel file)
- back to the list of vocabularies

No.	Column name	Data type	Notes
1	CODE	Varchar(5)	Primary key
2	DESCRIPTION	Varchar(20)	
3	STATUS	Char(1)	
4	TRANSLATION	Varchar(20)	
5	ORDER_CTRL	Numeric(22)	
6	DATE_ENTERED	Date	
7	DATE_UPDATED	Date	

Sample data (rows 1 to 7 of 7)

CODE	DESCRIPTION	STATUS	TRANSLATION	ORDER_CTRL	DATE_ENTERED	DATE_UPDATED
A	ABUNDANT	C	Abundant	5	04-JUN-03	
C	COMMON	C	Common	10	04-JUN-03	
R	RARE	C	Rare	15	04-JUN-03	
P	PRESENT	C	Present	20	04-JUN-03	
*	NOT APPLICABLE	C	Not Applicable	900	04-JUN-03	
!	NOT AVAILABLE	C	Not Available	905	04-JUN-03	
?	NOT ENTERED	C	Not Entered	910	04-JUN-03	

Figure 2.15. Screenshot of the BGS Vocabulary Web Service

4.2 *Comparison of Existing Semantic Frameworks*

4.2.1 *Comparison Criteria*

We will compare the semantic frameworks introduced above using two types of criteria: semantics capabilities and user interface capabilities. Semantics capabilities have to do with the ability of the semantic framework to perform semantics-based operations on data,

[44] 02 June 2010

[45] http://www.bgs.ac.uk/data/vocabularies/home.html

metadata or services. User interface capabilities deal with the functionalities provided to (human or machine) users.

Semantics Capabilities

- *Data Discovery* – ability to use semantic relationships (such as narrower, broader, related, etc.) between terms (metadata keywords) for improving data discovery; for instance if a user is searching for data using a given keyword (e.g. coastline), synonyms (e.g., shoreline) and narrower terms (e.g.) can be used to improve the completeness of the results;
- *Service Discovery* – ability to use semantic relationships between terms (metadata keywords) for improving service discovery;
- *Interoperability* – ability to perform semantic data, metadata or service interoperability based on semantic relationships between heterogeneous vocabularies or data structures;
- *Service Chaining* – ability to perform semantics-based service chaining; examples of this include the ability of checking the validity of a service chain from a structural and semantic point of view, and the ability of managing uncertainty propagation rules in service chaining;
- *Multi-Domain Support* – support for multi-domain ontologies; this is the case where several ontologies from different domains are mapped to each other and used to build the knowledge base; each ontology may be considered on its own or combined with the remaining ontologies;
- *Multilingual Support* – support for multi-lingual vocabularies or ontologies;
- *Multi-Facet browsing/search* – support for multi-facet data/service search or browsing for instance data/service search by theme, place, parameter or instrument;

User Interface Capabilities

- *Ontology Browsing* – support for ontology or vocabulary browsing by users;
- *Smart Search* – support for meaning-based free text search by users; for instance by comparing the text entered by the user to ontology terms and their definitions and returning the most relevant results to the users text;
- *Ontology Delivery* – ability for users to discover and download ontologies through the semantic framework;
- *Semantic Queries* – support for semantics-based queries by users;
- *Registering Ontologies* – ability of the users to submit and register their ontologies.

4.2.2 *Comparison*

Table 4-1 provides a comparison matrix for the semantic frameworks introduced in section 4.1 using the criteria defined in section 4.2.1.

Table 4-1 Comparison Matrix for the Reviewed Semantic Frameworks (X: supported, x: partly supported)

	Data Discovery	Service Discovery	Interoperability	Service Chaining	Multi-domain support	Multilingual Support	Multi-Facet Browsing/Search	Ontologies Browsing	Smart Search	Ontology Delivery	Semantic Queries	Registering Ontologies
OTEG	X				X			X	X			X
InterRisk	X				X			X	X			X
ICAN	X	X	X					X				
MMI			x		X					X	X	X
OOI CI			X				X	X		X	X	X
GEMET					X	X	X	X	x			
ENVISION & SWING	X	X	X	X				X	X	X	X	
ORCHESTRA	X	X	X									
OOSTethys	X	X	X									X
VSTO	X		x		X		X	x		x		
NASA GCMD	X				X			X	x			
USGS EWDC	X							X	x			
NERC & SeaDataNet	X		x	x	X		X	X	X	X	X	
BGS										X		

5 Semantics Requirements for the NETMAR Case Studies

The aim of this chapter is to identify the semantics requirements of each of the NETMAR case studies (c.f., section 5.1). Summary requirements are then compiled in section 5.2.

5.1 Requirements per Case Study

5.1.1 *CS-1: Arctic Sea Ice and Metocean Observing System*

The use case for comparing sea ice concentration and extent from different sources will benefit from use of ontologies. Specifically, to be able to perform:

- Smart search functionality for allowing users to search for available datasets using ontology terms and their relationships.
- Multi-facets data discovery based on a combination of multiple keywords, including parameter, instrument, sensor, themes, disciplines. (Any combination of these keywords should be searchable.)

It may also be useful to be able to perform:

- Semantic service discovery to find services that can process certain types of satellite data and estimate sea ice concentration and extent. Related to this, it will also be useful to be able to search for data that can be used as input to these services; again the ability to use a semantic data discovery service would be useful.

5.1.2 *CS-2: Near Real Time Monitoring and Forecasting of Oil Spill*

The use case Oil Spill management at sea will take benefit from the use of ontologies. The main requirements identified include:

- Data Discovery – ability to perform semantic data discovery
- Multi-Facet browsing/search – support for multi-facet data/service search or browsing
- Smart Search – support for meaning-based free text search by users

Other optional functionalities to integrate in the CS-2 include:

- Service Discovery – ability to perform semantic service discovery
- Interoperability – ability to perform semantic data, metadata or service interoperability
- Multilingual Support – support for multi-lingual vocabularies or ontologies

5.1.3 *CS-3: Relationships between physical and biological variables*

Researchers in ecosystems need to be able to identify and use long term time series in order to quantify ecosystem responses to natural variability, climate change or the impact of anthropogenic activities.
Operational users may find it useful to compare, in near real time, contemporary satellite and in situ data in order to provide input to water quality monitoring systems.

The following is the list of the semantic capabilities required by CS-3.

- *Data Discovery* – Researchers in ecosystems need to be able to identify useful datasets; for example by simple search, by parameter search or by compatibility with another dataset or service
- *Service Discovery* – Researchers in ecosystems need to be able to identify NETMAR services for working with these datasets as above

- *Interoperability* – There must be a means of translating between different parameter names where these refer to compatible data. For instance satellite-derived chlorophyll might be referred to by one name (chl-a) and an in situ measurement by another (chlorophyll by fluorescence).
- *Service Chaining* – Metadata defining the data and service parameters must be made available in such a way that a consumer (e.g. the service editor or the processing service itself) can verify that a particular dataset is a suitable input semantically for a service. We envisage that this metadata would be referenced (but not held) within the standard metadata for these objects (e.g. a related URL link to the semantic metadata within the GetCapabilities results). In addition metadata representing uncertainty/error for a dataset/service should be made available in the same way and be capable of propagating through the service chain.

5.1.4 *CS-4: Ecosystem Model Validation*

There is a need to compare the coupled physical and biological models which are being run in hindcast mode with historical EO data for validation, enabling future forecasts to observe the impacts of climate change such as changes in primary production or ocean acidification.
At present there is no automated way in which a scientist can select model output and matching EO data and compare them, each comparison must be set up manually. The semantic framework and the service chaining editor provided by NETMAR allow compatible datasets to be fed into a model/EO comparison engine and predefined statistical comparisons to be carried out.

CS-4 has the same semantic framework requirements as CS-3.

5.1.5 *CS-5: International Coastal Atlas Network (ICAN)*

The coastal atlas interoperability prototype, developed by CMRC as part of their involvement in ICAN, currently supports data discovery, catalogue services interoperability and basic ontology browsing. Further requirements to improve the prototype have been expressed by the ICAN community at the third and fourth ICAN workshops [DW08, WD10]. The main requirements identified include:

- Ontology-based semantic interoperability of OGC Web Feature Services (WFS) for sharing vector data;
- Multi-facets data discovery based on three types of metadata keywords: themes, instruments and disciplines;
- Advanced ontology browsing allowing for the visualisation of term definitions, related terms, and ontology graph browsing;
- Smart search functionality for allowing users to search ontology terms and data by meaning.

Other optional functionalities to integrate in the ICAN prototype include:

- Semantic service discovery;
- Support for multi-domain ontologies;
- Multilingual support.

5.1.6 *CS-6: Phytoplankton Blooms in gulf of Biscay and English Channel*

The use case about the monitoring of phytoplankton blooms in gulf of Biscay and English Channel will take benefit from the use of ontologies. The main requirements identified include:

- Data Discovery – ability to perform semantic data discovery
- Multi-Facet browsing/search – support for multi-facet data/service search or browsing

- Smart Search – support for meaning-based free text search by users

Other optional functionalities to integrate in the CS-6 include:
- Service Discovery – ability to perform semantic service discovery
- Interoperability – ability to perform semantic data, metadata or service interoperability
- Multilingual Support – support for multi-lingual vocabularies or ontologies

5.2 Summary of Requirements for the Case Studies

Table 5-1 summarises the requirements identified above for each case study using the semantics capabilities and user interface functionalities identified in Section 4.2.

Table 5-1 Summary of Requirements per Case Study (X: required, x: desirable)

	Data Discovery	Service Discovery	Interoperability	Service Chaining	Multi-domain support	Multilingual Support	Multi-Facet Browsing/Search	Ontologies Browsing	Smart Search	Ontology Delivery	Semantic Queries	Registering Ontologies
CS-1: Sea Ice	x	x					X		X			
CS-2: Oil Spills	X	x	x			x	X		X			
CS-3: Relationships…	X	X	X	X		x	X		X		X	
CS-4: Ecosystem	X	X	X	X		x	X		X		X	
CS-5: ICAN	X	x	X		x	x	X	X	X			
CS-6: Phytoplankton	X	x	x			x	X		X			

Requirements for the NETMAR semantic framework are summarised in Table 5-2. Support for multi-domain ontologies is needed given the variety of the case studies that belong to different domains.

Table 5-2 Requirements for the NETMAR Semantic Framework (X: required, x: desirable)

	Data Discovery	Service Discovery	Interoperability	Service Chaining	Multi-domain support	Multilingual Support	Multi-Facet Browsing/Search	Ontologies Browsing	Smart Search	Ontology Delivery	Semantic Queries	Registering Ontologies
NETMAR S.F.	X	X	X	X	X	x	X	X	X		X	

6 Recommendations

The aim of this chapter is to make recommendations for the NETMAR semantic framework capabilities and functionalities to be implemented, based on the existing EIS semantic frameworks reviewed in chapter 4 and on the case study requirements identified in chapter 5.

The recommendations for the NETMAR semantic framework are listed below.

REC.SF.12

The NETMAR semantic framework SHOULD build on existing semantic framework architectures such as ORCHESTRA, ICAN, MMI and SeaDataNet. More specifically it SHOULD use a semantic query processor for rewriting user queries using the semantic knowledge provided as part of the NETMAR ontologies. The semantic query processor will

1. Improve data and service discovery by ensuring complete results while searching data and service,
2. Help checking the semantic validity of user-defined service chains by checking the semantic compatibility of the data to input into a service against the service description.

REC.SF.13

Ontology browsing in the NETMAR semantic framework SHOULD use a similar approach than that of the OTEG and SeaDataNet semantic framework and SHOULD drive data discovery by providing the list of product types or products for each concept selected by the user.

PER.SF.5

A similar approach to that of NASA GCMD MAY be used for multi-facet search and browsing.

PER.SF.6

The NETMAR semantic framework MAY be physically connected to existing semantic frameworks such as SeaDataNet, MMI or GEMET.

REC.SF.14

The NETMAR Services SHOULD be described semantically in a service registry. Semantic descriptions of services will

1. Help semantic service discovery,
2. Help the service composition editor check the semantic validity of service chains built by users.

7 References

[BB08] Beckett, D., and Berners-Lee, T.: Turtle – Terse RDF Triple Language. W3C Team Submission. 14 January 2008. Available online at: http://www.w3.org/TeamSubmission/turtle/.

[Be09] Bermudez, L.: OOI Semantic Prototype. November 2009. Available online at: http://oceanobservatories.org/spaces/display/CIDev/Semantic+Framework+Integration.

[BG04] Brickley, D., and Guha, R.V.: RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation. 10 February 2004. Available online from: http://www.w3.org/TR/rdf-schema/.

[BM04] Beckett, D., and McBride, B.: RDF/XML Syntax Specification (Revised). W3C Recommendation. 10 February 2004. Available at: http://www.w3.org/TR/rdf-syntax-grammar/.

[BODC10] British Oceanographic Data Centre, 2010. Review of available ontology tooling. NETMAR Deliverable D3.2. In preparation.

[CTL09] Coene, Y., Truong-Minh, H. and Lassoued, Y., 2009. Discovery standards in HMA-T and Discovery in FP6 InterRisk. Presentation at Ontology and Discovery Workshop - ESRIN 4 March 2009. Available at: http://wiki.services.eoportal.org/tiki-download_wiki_attachment.php?attId=278&page=Ontology and Discovery Workshop - ESRIN 4 March 2009&download=y.

[DW08] Dwyer, N., and Wright, D.J.: Report of International Coastal Atlas Network Workshop 3 on Federated Coastal Atlases: Building on the Interoperable Approach, European Environment Agency, Copenhagen, Denmark, 7-11 July 2008.

[IS09] Isaac, A., and Summers, E.: SKOS Simple Knowledge Organization System Primer. W3C Working Group Note 18 August 2009. Available online at: http://www.w3.org/TR/skos-primer/.

[KS09] Klopfer, M.; and Simonis, I. (editors), 2009. SANY - an open service architecture for sensor networks. Available at http://sany-ip.eu/publications/3317.

[MH04] McGuinness, D., and Harmelen, F.: OWL Web Ontology Language: Overview. W3C Recommendation. 10 February 2004. URL: http://www.w3.org/TR/owl-features/.

[MGH09] Motik, B., Grau, B.C., Horrocks, I., Wu, Z., Fokoue, A., and Lutz, C.: OWL 2 Web Ontology Language Profiles. W3C Recommendation. 27 April 2009. Available online at: http://www.w3.org/TR/owl2-profiles/.

[MB09] Miles, A., and Bechhofer, S.: SKOS Simple Knowledge Organization System: Reference. W3C Recommendation. 18 August 2009. Available online at: http://www.w3.org/TR/2009/REC-skos-reference-20090818/.

[PH10] Perry, M., and Herring, J.: GeoSPARQL – A geographic query language for RDF data. A proposal for an OGC Draft Candidate Standard. Open Geospatial Consortium. 27 April 2010.

[PS08] Prud'hommeaux, E., and Seaborne, A.: SPARQL Query Language for RDF. W3C Recommendation. 15 January 2008. URL: http://www.w3.org/TR/rdf-sparql-query/.

[RDF04] RDF Working Group: Resources Description Framework, World Wide Web Consortium (W3C), 10 February 2004. Available at: http://www.w3.org/RDF/.

[Us07] Usländer, T.: Reference Model for the ORCHESTRA Architecture (RM-OA) V2 (Rev 2.1). Open Geospatial consortium Inc. 10 August 2007.

[W3C09] W3C OWL Working Group: OWL 2 Web Ontology Language: Document Overview. W3C Recommendation. 27 October 2009. Available online at: http://www.w3.org/TR/owl2-overview/.

[WD10] Wright, D.J., Dwyer, N., Kopke, K., and O'Dea, L.: Report of International Coastal Atlas Network Workshop 4: formalizing the Network, Engaging the Mediterranean, UNESCO International Centre for Theoretical Physics, Trieste, Italy, 16-20 November 2009.

8 Acronyms

API	Application Programming Interface
BGS	British Geological Survey
CEDAR	Coupling, Energetics and Dynamics of Atmospheric Regions
CF	Climate and Forecast
CMRC	Coastal and Marine Resources Centre
CS	Case Study
CSW	Catalogue Service for the Web
DIF	Directory Interchange Format
EEA	European Environmental Agency
EIS	Environmental Information System
EUMIS	European Marine Information System
ENVISION	Environmental Services Infrastructure with Ontologies
EO	Earth Observation
ETC/CDS	European Topic Centre on Catalogue of Data Sources
EWDC	Enterprise Web Document Catalog
FP6	Framework Programme 6
GCMD	Global Change Master Directory
GEMET	General Multilingual Environmental Thesaurus
GMES	Global Monitoring for Environment and Security
HSQLDB	Hyper Structured Query Language Database
HTTP	Hypertext Transfer Protocol
ICAN	International Coastal Atlas Network
ICT	Information and Communication Technology
InterRisk	Interoperable GMES Services for Environmental Risk
IRI	Internationalized Resource Identifier
JDBC	Java Database Connectivity
MIDA	Marine Irish Digital Atlas
MLSO	Mauna Loa Solar Observatory
MMI	Marine Metadata Interoperability
NASA	National Aeronautics and Space Administration
NERC	Natural Environmental Research Council
NETMAR	Open service Network for Marine Environmental Data
NVS	NERC Vocabulary Server
OCA	Oregon Coastal Atlas
OGC	Open Geospatial Consortium
OOI	Ocean Observatories Initiative
OOI CI	OOI Cyberinfrastructure

OPeNDAP	Open-source Project for a Network Data Access Protocol
ORCHESTRA	Open Architecture and Spatial Data Infrastructure for Risk Management
OTEG	Open Access Ontology/Terminology for the GMES Space Component
OWL	Web Ontology Language
PHP	PHP Hypertext Preprocessor *(Where: PHP = Personal Home Page)*
RAP	RDF API for PHP
RDBMS	Relational Database Management System
RDF	Resource Description Framework
RDFS	(Also RDF-S) RDF Schema
ReST	(Also REST) Representational State Transfer
SKOS	Simple Knowledge Organisation System
SOFA	Simple Ontology Framework API
SPARQL	SPARQL Protocol and RDF Query Language *(Recursive acronym)*
SSTSP	solar, solar-terrestrial, and space physics
SWING	Semantic Web services Interoperability foe Geospatial decision making
SWSWiki	Semantic Web Standards Wiki
Turtle	Terse RDF Triple Language
URI	Uniform Resource Identifier
USGS	US Geological Survey
VSTO	Virtual Solar Terrestrial Observatory
W3C	World Wide Web Consortium
WCS	Web Coverage Service
WFS	Web Feature Service
WMS	Web Map Service
WSMO	Web Service Modelling Ontology
WSMX	Web Service Modelling execution
XML	eXtensible Markup Language

Appendices

Appendix A. List of Recommendations for the NETMAR Semantic Framework

RUL.SF.1

The NETMAR ontologies (but not thesauri or bridging mappings between semantic resources) SHALL be defined using a standard ontology language such as RDF/RDFS or OWL in order to facilitate interoperability with external standards-based semantic frameworks and ontologies.

REC.SF.1

Compared to RDF and RDFS, OWL offers more expressiveness and machine interpretability and allows for rich semantic relationships and rules. Therefore, the NETMAR ontologies SHOULD be implemented in OWL rather than RDF.

REC.SF.2

IF OWL is to be used as the ontology language for the NETMAR semantic framework, THEN, in order to cope with the evolution of the W3C Web Ontology Language and to profit from the new features of OWL 2, the NETMAR ontologies SHOULD be defined in OWL 2 rather than OWL 1. It is important to note here that most popular ontology tools and software already started supporting OWL 2.

RUL.SF.2

IF OWL is to be used as the ontology language for the NETMAR semantic framework, THEN OWL DL SHALL be used rather than OWL Lite or OWL Full. This would guarantee maximum expressiveness while retaining computational completeness and decidability.

REC.SF.3

IF, following REC.SF.1, REC.SF.1, and RUL.SF.2, OWL 2 DL is to be used as the ontology language in NETMAR, THEN an appropriate profile (i.e., OWL 2 EL, OWL 2 QL, or OWL 2 RL) SHOULD be carefully chosen depending on the type of application to be implemented. The use of OWL 2 profiles will ensure that reasoning be done in polynomial time. Moreover, the choice of the appropriate profile will ensure optimum balance of expressiveness vs. performance.

REC.SF.4

The SKOS model SHOULD be used for encoding thesauri and as a mechanism for specifying semantic relationships among concepts of the NETMAR ontologies and possibly relationships with external ontologies (if required). This would ensure that semantic relationships be implemented in a standard way in accordance with the W3C recommendations, which would facilitate interoperability with other ontologies and semantic frameworks.

RUL.SF.3

IF, according to REC.SF.4, SKOS is to be used in the NETMAR ontologies, THEN, according to the OWL DL restrictions, concepts and concepts schemes SHALL (i.e., must) be implemented only as instances of skos:Concept and skos:ConceptSchemes and not as owl:Class.

PER.SF.1

IF, when using SKOS with OWL DL, any concepts or concept schemes need to be defined as classes, THEN a workaround MAY be used. For instance one MAY define a separate class and an instance corresponding to the same concept or concept scheme, and link them using an owl:annotation or a restriction on the class instances.

REC.SF.5

XML is commonly accepted as the data format for sharing information on the web. Therefore, RDF/XML SHOULD be used as an exchange format for the NETMAR ontologies rather than Turtle.

PER.SF.2

The NETMAR ontologies MAY be queried, internally, by the NETMAR semantic framework using the SPARQL protocol.

REC.SF.6

IF the NETMAR ontologies are to be made publicly available THEN they SHOULD be delivered through an ontology server using a standard protocol such as SPARQL. This would allow external users to perform semantic queries on the NETMAR ontologies in a standard way and to reuse these in their applications.

REC.SF.7

Although GeoSPARQL is not yet a recommendation, the NETMAR community SHOULD keep an eye on the progress of this work and SHOULD consider actively participating in the development of this standard.

PER.SF.3

GeoSPARQL and its feature model MAY be used in NETMAR for representing and querying metadata place keywords, which MAY be enriched using geospatial properties such as geographic location or bounding box information.

REC.SF.8

Given that Jena and Sesame are the most established semantic frameworks and given the various features they provide (such as RDF stores, APIs, support for OWL and SPARQL) the NETMAR semantic framework SHOULD be developed in Java using Jena or Sesame.
More specifically, the authors recommend that Jena SHOULD be used rather than Sesame as the former provides an OWL view of ontologies, which is very handy when handling OWL ontologies.

REC.SF.9

Although Jena and Sesame can communicate with each other using the Jena Sesame Model or the Sesame-Jena adaptor, for sake of ease of implementation and maintenance, one SHOULD use the Jena API to access Jena databases and the Sesame API to access Sesame databases.

REC.SF.10

Scalability of the ontology base is an important issue that could be solved using RDF stores or relational databases. Therefore, the NETMAR ontologies SHOULD be stored and managed in an RDF store or a relational database.

REC.SF.11

The NETMAR semantic framework SHOULD implement a customised ontology browser for displaying useful concept graphs based on the SKOS semantic relationships. Examples of such graphs include the hierarchy of concepts using the `skos:narrower` and `skos:broader` relationships, and the graph of semantically related terms.

PER.SF.4

Whenever suitable, jOWL component MAY be adapted and reused to build the NETMAR ontology browser. Other standard graph libraries such as JavaScript, Adobe Flex or JavaFX graph libraries (trees, graphs, etc.) MAY also be used to build the NETMAR ontology browser.

REC.SF.12

The NETMAR semantic framework SHOULD build on existing semantic framework architectures such as ORCHESTRA, ICAN, MMI and SeaDataNet. More specifically it SHOULD use a semantic query processor for rewriting user queries using the semantic knowledge provided as part of the NETMAR ontologies. The semantic query processor will

1. Improve data and service discovery by ensuring complete results while searching data and service,
2. Help checking the semantic validity of user-defined service chains by checking the semantic compatibility of the data to input into a service against the service description.

REC.SF.13

Ontology browsing in the NETMAR semantic framework SHOULD use a similar approach than that of the OTEG and SeaDataNet semantic framework and SHOULD drive data discovery by providing the list of product types or products for each concept selected by the user.

PER.SF.5

A similar approach to that of NASA GCMD MAY be used for multi-facet search and browsing.

PER.SF.6

The NETMAR semantic framework MAY be physically connected to existing semantic frameworks such as SeaDataNet, MMI or GEMET.

REC.SF.14

The NETMAR Services SHOULD be described semantically in a service registry. Semantic descriptions of services will

1. Help semantic service discovery,
2. Help the service composition editor check the semantic validity of service chains built by users.