



Project No. 249024

NETMAR

Open service network for marine environmental data

Instrument: <i>Please tick</i>	CA	<input checked="" type="checkbox"/> STREP	IP	NOE
--	----	--	----	-----

ICT - Information and Communication Technologies Theme

D6.4 Second version of EUMIS subsystems

Reference: D6.4_Second_version_EUMIS_subsystems_r1_20120731

Due date of deliverable (as in Annex 1): M0 + 30

Actual submission date: 31 July 2012

Start date of project: 1 February 2010

Duration: 3 years

Nansen Environmental and Remote Sensing Center

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 55 20 58 00 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. Øystein Torgert (oysteint@met.no)

Author(s)

- Torill Hamre (torill.hamre@nersc.no)
- Yassine Lassued (y.lassoued@ucc.ie)
- Peter Walker (petwa@pml.ac.uk)
- Jorge de Jesus (jmdj@pml.ac.uk)
- Mickael Treguer (mickael.treguer@ifremer.fr)

Document approval

- Document status: Revision 1
- WP leader approval: 2012-07-31
- Quality Manager approval: 2012-07-31
- Coordinator approval: 2012-07-31

Revision History

Issue	Date	Change records	Author(s)
Draft	2012-06-15	Topic layout.	T. Hamre
Draft	2012-06-20	Updated description of search client.	Y. Lassoued
Draft	2012-07-10	Updated description of the service chaining editor.	P. Walker
Draft	2012-07-18	Updated description of the GIS viewer.	M. Treguer
Draft	2012-07-20	Updated the executive summary and conclusion.	T. Hamre
Draft	2012-07-31	Incorporated comments from quality control.	T. Hamre
1	2012-07-31	Final version approved by coordinator.	T. Hamre

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. The semantic tools will be implemented as portal components (called portlets) and web services. Portlets for retrieval and display of geographic datasets, “smart” search and discovery, and service chain composition will be developed and integrated in the EUMIS portal. This report describes the second version of these three portlets:

1. GIS map viewer
2. Smart search and discovery client
3. Service chaining editor

The **GIS map viewer** is able to retrieve map layers from a network of data servers, through the WMS (Web Map Service) standard protocol. Which WMS servers to connect to, can be configured dynamically while the application is running. Layers from the connected WMS servers are shown in a menu, and the user can choose which to display in the map window. Common GIS operations like zooming, panning and restoring the map back to the original extent are implemented. For layers such as satellite images the user can modify the colours used when displaying it in the map window. For layers containing in situ observations as a time series of parameter values, the user can click on the symbol representing the sensor and get the full list of measurements at that station. The GIS map viewer supports multiple map projections and also allows execution of a remote WPS service.

The **smart search and discovery client**, also referred to as the **semantic discovery client**, has two main features:

1. Ontology browsing: You can discover and navigate through and across the hierarchies of concepts defined in the NETMAR ontologies (thesauri). You can start from broad concepts in one thesaurus and navigate to more specific terms or related terms within or outside of that thesaurus. You can also search for terms in one or more thesauri. For each term, its definition is given, and you can easily navigate to narrower, broader and related terms, before selecting the term (topic) of interest and find associated datasets or services.
2. Faceted smart data and service search: You can search for resources (products and/or services) according to different facets (topic categories, e.g., Theme, Parameter, Instrument, Platform, etc.) or by keyword. “Smart search” refers to the ability of searching for resources by meaning rather than by simple textual match. Data search results may be geographically filtered through bounding boxes, drawn on a map or obtained from an ontology of place names.

The **service chaining editor** provides a graphical user interface (GUI) for creating and editing a workflow by selecting and connecting icons that represent a data delivery service or a data processing service. The workflow is stored and can be loaded again later for editing and updating. The output workflow can be executed by a workflow engine like Taverna.

All three components are developed using open source libraries, such as MapFaces and OpenLayers, which have an active development community, are widely used by web-GIS developers and that can be integrated in an open source Java application server. For the underlying infrastructure, the Liferay Community Edition has been selected. The components will be integrated in the EUMIS portal and used in the final user demonstrations, to be held during autumn 2012. They provide advanced semantic search and discovery, map display of both premade and dynamically generated products, as well as easy to use service chaining facilities, and will allow a full demonstration of the semantic tools and EUMIS portal solution that are being developed in the NETMAR project.

Contents

EXECUTIVE SUMMARY	II
1 INTRODUCTION.....	1
1.1 BACKGROUND	1
1.2 OBJECTIVE OF THIS REPORT	3
1.3 TERMINOLOGY	3
1.4 ABBREVIATIONS	4
1.5 ORGANISATION OF THIS REPORT	4
2 GIS MAP VIEWER.....	5
2.1 OBJECTIVES AND FEATURES	5
2.2 DESIGN AND IMPLEMENTATION ISSUES	5
2.2.1 <i>Design</i>	5
2.2.2 <i>Functionalities</i>	6
2.3 EXAMPLES OF USE	8
3 SMART SEARCH AND DISCOVERY CLIENT	10
3.1 OBJECTIVES AND FEATURES	10
3.2 DESIGN AND IMPLEMENTATION ISSUES	10
3.2.1 <i>Design</i>	10
3.2.2 <i>Use Case 1 – Ontology Browsing</i>	12
3.2.3 <i>Use Case 2 – Data and Service Search</i>	14
3.2.4 <i>Implementation Issues</i>	15
3.3 EXAMPLES OF USE	15
4 SERVICE CHAINING EDITOR	20
4.1 OBJECTIVES AND FEATURES	20
4.2 DESIGN AND IMPLEMENTATION ISSUES	20
4.2.1 <i>Client side editor GUI</i>	20
4.2.2 <i>WSDL support</i>	21
4.2.3 <i>Workspace, see Figure 4-2</i>	21
4.2.4 <i>Workflow export, see Error! Reference source not found.</i>	22
4.2.5 <i>Zoom control</i>	22
4.3 USAGE EXAMPLES	23
4.3.1 <i>Combining Satellite and Model Data</i>	23
4.3.2 <i>Simple WebGIS facilities</i>	24
4.3.3 <i>Workflow execution</i>	24
5 CONCLUSIONS	26
6 REFERENCES	27

1 Introduction

1.1 Background

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. The semantic tools will be implemented as portal components (called portlets) and web services. Portlets for retrieval and display of geographic datasets, “smart” search and discovery, and service chain composition have been developed for integration in the EUMIS portal.

The data delivery and processing services available in EUMIS will come from a distributed network of services, as illustrated in Figure 1-1. Data products observed from satellites, aircrafts, helicopters, vessels as well as buoys and other instrumented in the ocean, and products derived from these sources will be made as data delivery services through standard OGC WxS or OPeNDAP interfaces. Model predictions will also be made available through the service network using the same standard interfaces.

Measured and derived products can be used to generate products on the fly, by means of data processing services, which often combines data from two types of sources, e.g. satellite and aircraft observations of a potential oil spill to generate a new product. A processing service can also generate a new product based on a single data source alone, e.g. computing statistics from time series of satellite data. All data delivery and processing services will be available in EUMIS, and new workflows to compute customised products can be defined using the service chaining editor for later execution in a workflow engine like Taverna. The user can draw upon the smart search facilities of EUMIS to identify services of interest, and the retrieved products, both the pre-made from delivery services and those generated dynamically through processing services can be displayed together in the GIS map viewer.

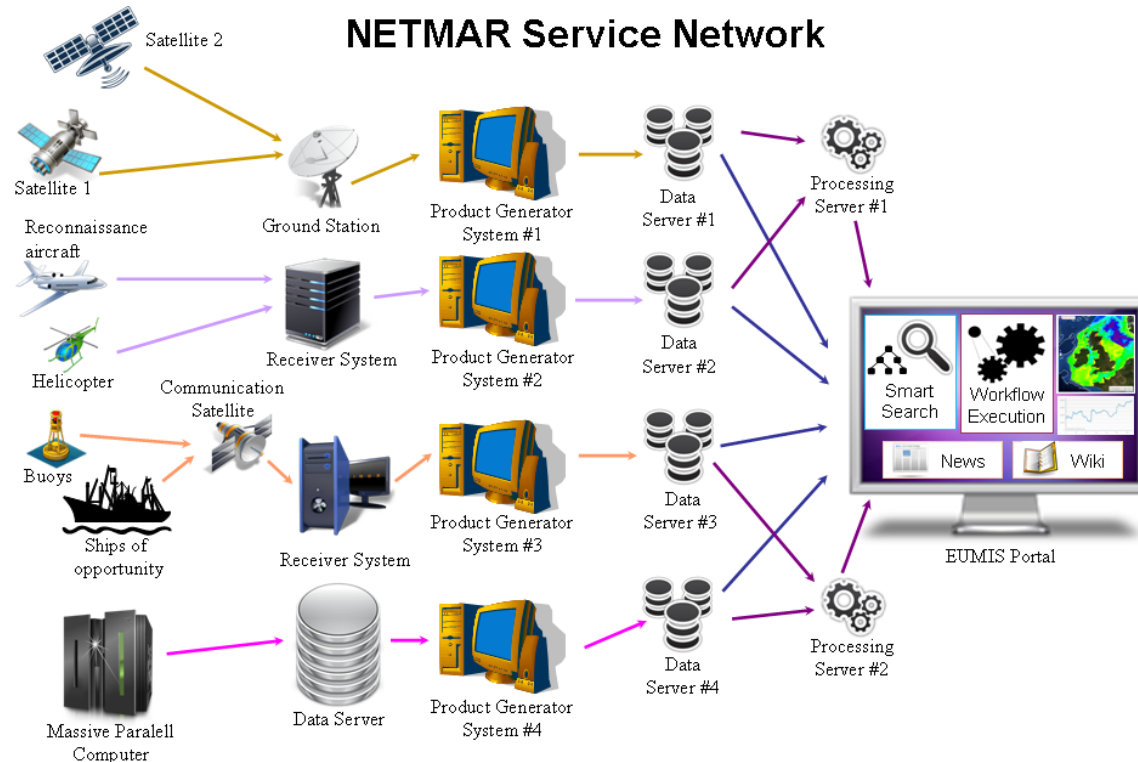


Figure 1-1 The NETMAR Service Network.

The EUMIS portal will also contain a Wiki, where users can find information about products and services, as well as technologies and tools that can be used to develop new services. EUMIS may also offer a News section and a Forum where registered users can discuss topics of interest related to these, respectively, depending on user interest in these features. These three components, Wiki, RSS news feed and Forum are all realised using built-in portlets in the Liferay Community Edition [LRv6], and therefore not included in this report.

For the NETMAR project, we have defined four pilots:

- Pilot 1: Arctic Sea Ice and Met-ocean Observing System
- Pilot 2: Oil spill drift forecast and Shoreline Cleanup assessment services in France
- Pilot 3: Ocean Colour – Marine Ecosystem, Research and Monitoring.
- Pilot 4: The International Coastal Atlas Network (ICAN)

The Arctic Ocean and the adjacent sub-Arctic seas, including Europe's northern frontiers, have become more important as a result of the global warming and increased economic activities related to energy exploration, marine resources and transportation. Pilot 1 is therefore addressing the need to access sea ice information and combine the information into an overall picture and assessment of historical, today's or predicted sea ice situations. Specifically the pilot is focused on how to: (1) use the EUMIS service composition editor to create service chains with data and processing services from different providers, and (2) use the EUMIS search interface to gain access to data and processing services provided by different by providers. The resulting ice products will be offered to representatives from offshore and shipping companies, ship and icebreaker captains/ice pilots, national authorities, regional environmental agencies, national ice services and scientists.

Marine oil pollution caused by accidents or illicit discharge from vessels can cause a severe and long term negative impact on the marine and coastal zone environment. Pilot 2 thus has focus on: (1) developing consensus forecasts for oil spill drift for improving the operational oil spill monitoring and forecasting service of France, and (2) providing access to all operational information about onshore pollution landings and cleanup site evolution in order to make them readily available during the crisis forecast service in France. The developed products are aimed at organisations and authorities involved in the operational oil spill monitoring and forecasting service of France.

Pilot 3 combines a number of related areas based around local, remote and modelled observations of biogeochemical parameters such as chlorophyll. Analysis and comparison of these data is an essential part of the work in the ocean colour community, and underlies most scientific and management activities – e.g. monitoring of algal blooms, validation of satellite sensors and prediction of future spread of a bloom. Thus, Pilot 3 focuses on the comparison of in-situ or model data with satellite data. Some of these activities operate within tight time constraints (e.g. monitoring) and require near-real time data and rapid processing, while others are longer term analyses. The developed products are aimed at the marine ecosystem research community and French authorities for coastal zone monitoring.

ICAN (International Coastal Atlas Network) [ICAN] is an informal group of organisations that have been meeting since 2006 to scope and implement data interoperability approaches to coastal web atlases (CWAs). In recent years significant momentum has occurred in the development of Internet resources for decision makers, scientists and the general public who are interested in the coast. To support these activities, Pilot 4 focuses on tools and ontologies for smart search and on making CWAs interoperable. The targeted users are members of the ICAN community.

The EUMIS components were designed and implemented to fulfil the requirements of these pilots, as identified through consultations with users in the respective domains about their requirements. The identified requirements were captured in scenario and use case descriptions, requirements lists and dedicated questionnaires [PTS+11].

1.2 Objective of this report

The purpose of this document is to describe the second version of the three EUMIS components: (1) the GIS map-viewer, (2) the semantic search and discovery client, and (3) the service chaining editor. These components will be used to demonstrate proof-of-concept of the semantic tools and the portal during autumn 2012.

1.3 Terminology

A **concept URI** is a pre-defined URI uniquely identifying a term (i.e. **concept**) in a vocabulary (i.e. controlled list of terms) for a given application domain. For example, the concept URI <http://vocab.ndg.nerc.ac.uk/term/P011/current/ATEMP2MM> identifies the term “Temperature (2m) of the atmosphere by model prediction” in the BODC Parameter Usage Vocabulary P011.

When describing a domain such as marine environmental sciences, it is common to think about things such as the instrument used to make measurements; the parameters measured; or where these measurements were taken. These groupings can have different levels of description within them. For example, thinking about measured parameters may lead to the idea of physical oceanographic measurements, which may lead to the idea of measuring salinity or of measuring temperature. These groupings may be called “**facets**”. The NETMAR semantic resource consists of the facets “parameter”, “instrument”, “platform (or platform class)”, “project” and “vertical coverage” [LL11].

A **portal** is a web site that collects input from a number of sources, and presents it in a uniform manner to the user. The portal content is perceived to come from the same source – the portal – while it typically is a combination of content from several sources, or an extract of selected content from a single external source (such as a news feed).

A **portlet** is portal component that can be deployed in a portal. A portlet can provide many types of functionality, among others, retrieve data from external sources, process and analyse data, present retrieved data on a geographic map. A portlet can also communicate with other portlets running in the same portal.

The **Simple Knowledge Organization System (SKOS)** is a W3C recommendation that provides a scheme for the serialisation of controlled vocabularies and thesauri [LL11]. SKOS is based upon **concepts** that it defines as a “unit of thought”, i.e. an idea or notion such as “shoreline emergency access” or “oil spill”. Concepts may also carry other information, e.g. relationships to other concepts and information about their provenance and version history.

A **Uniform Resource Identifier (URI)** is “a compact string of characters for identifying an abstract or physical resource” [RFC2396]. A URI can among others identify a file or a part of an online document, which can be identified by e.g. <ftp://ftp.is.co.za/rfc/rfc1808.txt> or <http://www.w3.org/TR/2009/REC-skos-reference-20090818/#concepts>.

A **vocabulary** can be either a list of terms or a list of terms and some text providing a definition of the term [LL11]. A vocabulary ensures that terms are used, and spelt, consistently. A vocabulary can be extended in its power by providing definitions of concepts.

1.4 Abbreviations

BPEL	Business Process Execution Language
CSW	Catalogue Services for the Web
CSWM	CSW (Catalogue Services for the Web) Mediator
EUMIS	European Marine Information System
GIS	Geographical Information System
GML	Geography Markup Language
GUI	Graphical User Interface
HTML	HyperText Markup Language
ICAN	International Coastal Atlas Network
JSF	Java Server Faces
JSON	JavaScript Object Notation
JSP	Java Server Pages
KML	Keyhole Markup Language
NETMAR	Open service network for marine environmental data
OPeNDAP	Open-source Project for a Network Data Access Protocol
SKOS	Simple Knowledge Organization System
SOAP	Simple Object Access Protocol
SWS	Semantic Web Service
URI	Uniform Resource Identifier
WCS	Web Coverage Service
WFS	Web Feature Service
WMS	Web Map Service
WPS	Web Processing Service
WSDL	Web Service Definition Language
WYSIWYG	What You See Is What You Get
XML	eXtensible Markup Language

1.5 Organisation of this report

The remainder of this report is organised as follows. Section 2 describes the GIS map viewer component. Section 3 describes the semantic search and discovery client, and Section 4 describes the service chaining editor. A short conclusion is found in the last section.

2 GIS map viewer

2.1 Objectives and features

The GIS Viewer is a component in the EUMIS portal that allows users to view distributed geo-spatial resources that have been shared by means of the OGC WMS standard [WMS13]. It provides a user-friendly way of browsing WMS servers to see what features they have available (GetCapabilities), choosing those of interest, and seeing the resulting map (GetMap). The GIS viewer allows the user to obtain multiple layers overlaid on top of one another in the same geographic map projection. And, if the WMS server supports it, it's possible to click on a feature to get additional information about it (GetFeatureInfo).

This component allows EUMIS users to visualise a large number of cartographic layers obtained from a network of WMS services. These services are provided by various software tools, such as MapServer [MS], Geoserver [GS], Constellation-SDI [CSDI], ncWMS [NCW12] and Thredds Data Server with ncWMS integrated [TDS12]. The GIS map viewer includes common GIS functions (e.g. zoom, pan, legends), with the addition of its capacity to interrogate multi-dimensional services, e.g. time and elevation.

The GIS viewer will be used to visualise layers in all four EUMIS pilots [PTS+11]:

- Pilot 1: Arctic Sea Ice and Met-ocean Observing System
- Pilot 2: Oil spill drift forecast and Shoreline Cleanup assessment services in France
- Pilot 3: Ocean Colour – Marine Ecosystem, Research and Monitoring.
- Pilot 4: The International Coastal Atlas Network (ICAN)

It will provide the basic GIS functionality for displaying raster and vector layers obtained from the EUMIS service network. This includes both layers retrieved from the data delivery services as well as layers generated dynamically as a result of executing a service chain (i.e. a series of processing services).

2.2 Design and implementation issues

2.2.1 Design

The GIS Map Viewer is written using the MapFaces library [MF]. MapFaces has a modular design consisting separate JSF (Java Server Faces) components resting on a base geo-spatial library. MapFaces consists of a server which services requests for the web application and handles asynchronous events generated by the application either by making appropriate calls to the base geo-spatial library, by making service requests to remote OGC compliant servers, or by handling needs of the user interface itself. The web application is sent to a client browser upon initial request, uses a combination of HTML and JavaScript [Fla11] to make asynchronous calls to the MapFaces server or make image requests directly to remote WMS servers (Figure 2-1).

The components and widgets follow the JSF architecture quite closely. They provide a JSF renderer element to generate the HTML and JavaScript required to recreate the user interface element in the client browser and handler code to dispatch events to the appropriate logic handling block in the code base.

The components rely on several JavaScript elements for user interface elements running in the browser on the client side. Mapfaces uses the jQuery javascript library to provide elegant and powerful components in a compact, highly structured library [JQ]. MapFaces also uses a simplified version of the OpenLayers JavaScript library for displaying map data in web browsers, providing functionality like transparent layers and simple navigation panel [OL].

The GIS viewer component is integrated into a portlet and tested in a Liferay Community Edition portal at Ifremer. This means it is ready for integration in the EUMIS portal, which is using the same version of the Liferay portal framework.

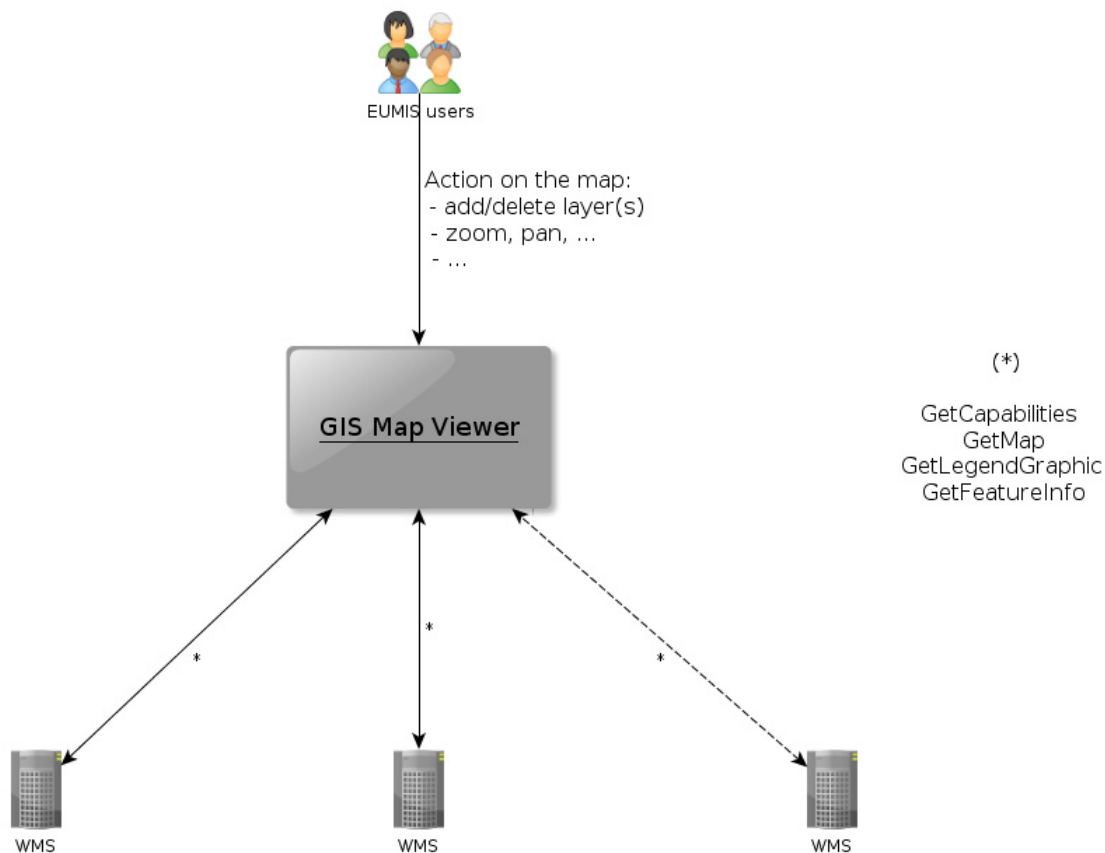


Figure 2-1 Protocol diagram for the GIS Map Viewer.

2.2.2 Functionalities

The main features of the GIS Map Viewer include:

- Common GIS functions (e.g. zoom, pan, legends, locator map, store the context map, print the map)
- NcWMS compatibility
- Multiple projections
- WPS (Web Processing Service¹) service execution

NcWMS is a Web Map Service for geospatial data that are stored in NetCDF files, like model output and satellite datasets. It implements some extensions to the standard, timeseries and transects. The GIS viewer is compliant with ncWMS, end users can interrogate multi-dimensional services (e.g. time and elevation), change the color map, graph timeseries and transects (Figure 2-2).

The GIS viewer is able to display geospatial layers into several map projections. If the WMS server can respond in the same projection of the map, the client part of the GIS viewer displays directly the layer, otherwise it's the server part of the GIS viewer that does the re-projection. So, the NETMAR pilots can use different map projections. For example, Pilot 1 (Arctic Sea Ice and Met-ocean Observing System) uses the stereographic projection (see Figure 2-3), while other pilots use the geographic latitude/longitude projection.

¹ <http://www.opengeospatial.org/standards/wps>

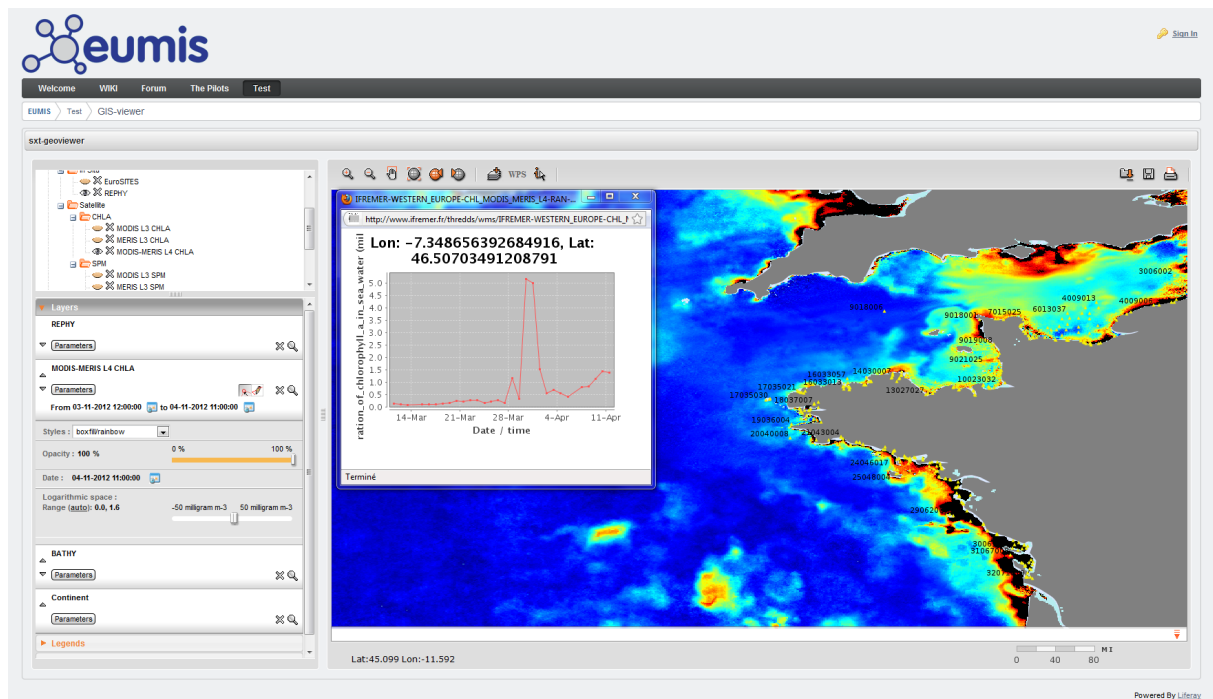


Figure 2-2 Example of timeseries from ncWMS.

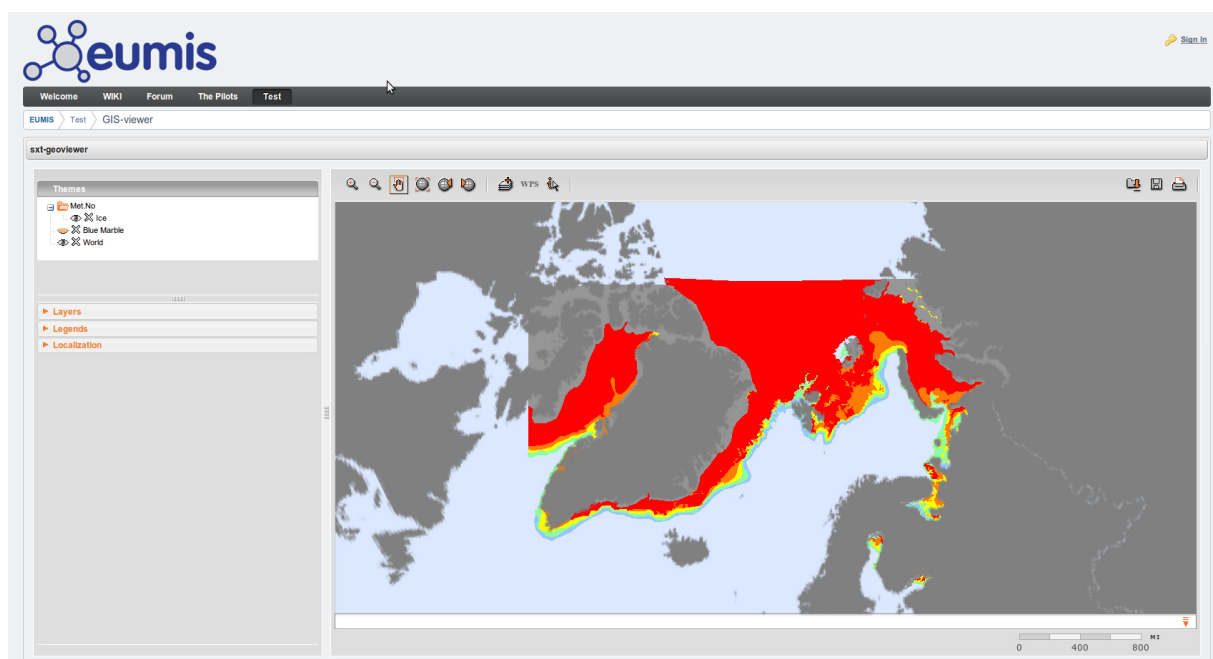


Figure 2-3 Ice chart in stereographic projection.

A WPS client is implemented in the GIS Viewer. It allows to execute remote WPS server, and when the result is a geospatial dataset (e.g. GeoTIFF, GML, KML) to display it in the map. The users can select or edit WPS URL, then choose an available Process, and set the input parameters, and finally execute the process (Figure 2-4).

The figure displays two side-by-side windows. The left window is a WPS form for the 'v.random' process. It contains several input fields and sections:

- 'Value (float)': empty
- 'Number of points to be created': input field with '10' entered.
- 'Reference': http://www.w3.org/TR/xmlschema-2/#integer
- 'Value (integer)': empty
- 'The seed to initialize the random generator. If not set the process will use a random seed': input field with 'http://www.w3.org/TR/xmlschema-2/#int' entered.
- 'Value (integer)': empty
- 'Create 3D output (optional)': dropdown menu with 'None' selected.
- '4 - Choose outputs': a section with a dropdown for 'Name for output vector map' and checkboxes for 'Enable' and 'By reference'.
- '5 - Execute the process': a section with an 'Execute' button.

The right window shows the XML response from the WPS server. It includes process metadata, status, and output data. The XML is structured as follows:


```

    <wps:ExecuteResponse xmlns:wps="http://www.opengis.net/wps" xmlns:ows="http://www.opengis.net/ows" xmlns:gml="http://www.opengis.net/gml" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.opengis.net/wps http://schemas.opengis.net/wps/2.0.0/wps.xsd">
      <wps:Process wps:processVersion="1">
        <ows:Identifier>v.random</ows:Identifier>
        <ows:Title>Generates randomly 2D/3D vector points map.</ows:Title>
        <ows:Abstract>http://grass.osgeo.org/grass70/manuals/html70_user/v.random.html</ows:Abstract>
        <ows:Metadata>
          <ows:Metadata xmlns:xlink="http://www.w3.org/1999/xlink" xlink:href="http://www.opengis.net/wps/2.0.0/wps.xsd" xlink:title="statistics" xlink:href="http://www.opengis.net/wps/2.0.0/wps.xsd" />
          <ows:Metadata xmlns:xlink="http://www.w3.org/1999/xlink" xlink:href="http://www.opengis.net/wps/2.0.0/wps.xsd" />
        </ows:Metadata>
        <ows:Status>
          <ows:Status creationTime="2012-04-18T22:17:54Z">
            <ows:ProcessSucceeded>PyWPS Process v.random successfully calculated</ows:ProcessSucceeded>
          </ows:Status>
        </ows:Status>
        <ows:ProcessOutputs>
          <ows:Output>
            <ows:Identifier>output_0c9GHiv.xsd</ows:Identifier>
            <ows:Title>Name for output vector map</ows:Title>
            <ows:Data>
              <wps:ComplexData schema="http://schemas.opengis.net/gml/3.1.1/base/gml.xsd" encoding="UTF-8">
                <ogr:FeatureCollection xsi:schemaLocation="http://ogr.maptools.org/ output_0c9GHiv.xsd">
                  <gml:boundedBy>
                    <gml:Box>
                      <gml:coord>
                        <gml:X>0.2178626136006159</gml:X>
                        <gml:Y>9.52477599987691</gml:Y>
                      </gml:coord>
                      <gml:coord>
                        <gml:X>97.67564157847112</gml:X>
                        <gml:Y>98.24098348535642</gml:Y>
                      </gml:coord>
                    </gml:Box>
                  </gml:boundedBy>
                  <gml:featureMember>
                    <ogr:output fid="F0">
                      <ogr:geometryProperty>
                        <gml:Point>
                          <gml:coordinates>76.96700570032327,21.22931257878864,0</gml:coordinates>
                        </gml:Point>
                      </ogr:geometryProperty>
                      <ogr:cat>1</ogr:cat>
                    </ogr:output>
                  </gml:featureMember>
                  <gml:featureMember>
                    <ogr:output fid="F1">
                      <ogr:geometryProperty>
                        <gml:Point>
                          <gml:coordinates>0.217862613600616,93.413071750389904,0</gml:coordinates>
                        </gml:Point>
                      </ogr:geometryProperty>
                      <ogr:cat>2</ogr:cat>
                    </ogr:output>
                  </gml:featureMember>
                </ogr:FeatureCollection>
              </wps:ComplexData>
            </ows:Data>
          </ows:Output>
        </ows:ProcessOutputs>
      </wps:Process>
    </wps:ExecuteResponse>
  
```

Figure 2-4 WPS form (left) and response (right).

2.3 Examples of use

Figure 2-5 shows an example of an overlay of a raster layer and a vector layer. The raster layer contains a chlorophyll-a concentration image derived from satellite data, and the vector layer contains the location and data of observing stations in the French REPHY coastal monitoring network. The raster layer is displayed using the colour coding defined by the WMS server providing the data, and the vector layer is displayed with symbols (yellow triangles) marking the position of each station. In Figure 2-5, also the station numbers of some of the stations are shown.

Different satellite images can be selected by sliding the date selector in the GIS viewer layer list (left part of Figure 2-6), and the colours used when displaying the currently selected image can be modified interactively. The GIS viewer also shows a legend showing the correlation between the colour coding and satellite-derived chlorophyll-a values. For raster layers, it is also possible to modify the opacity of the image displayed. This is useful when overlaying multiple raster layers, as it allows the users to “see through” the layer displayed on top of the underlying layer(s). For the vector layer, the user can obtain the measured values by clicking on any one of the REPHY stations. The measured values are then retrieved from the WMS server (which supports GetFeatureInfo) and shown inside the map viewer window (lower part of Figure 2-6).

The GIS map viewer also provides information about the current position (latitude, longitude) of the cursor when inside the map, and the map scale. This information is shown at the bottom of both Figure 2-5 and Figure 2-6.

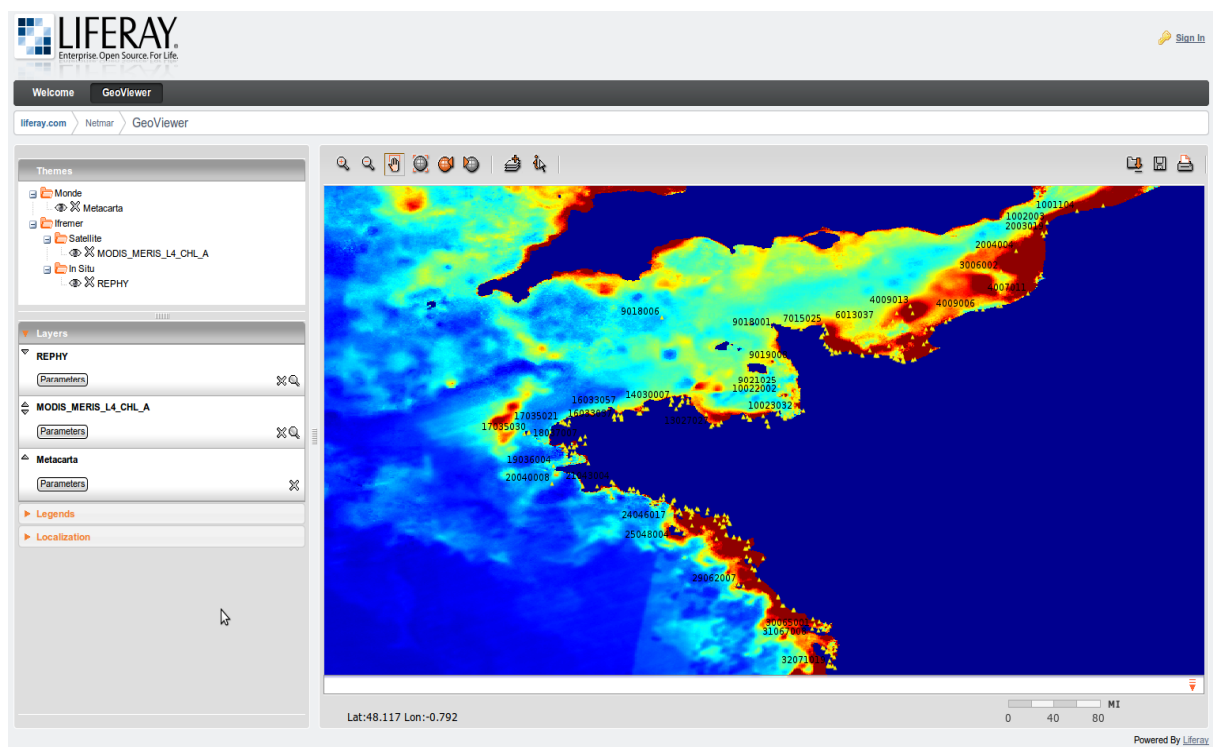


Figure 2-5 Example of satellite (Chl-a from MODIS-MERIS) and In situ (Phytoplankton and Phytotoxins networks : REPHY) dataset displayed in the GIS map viewer.

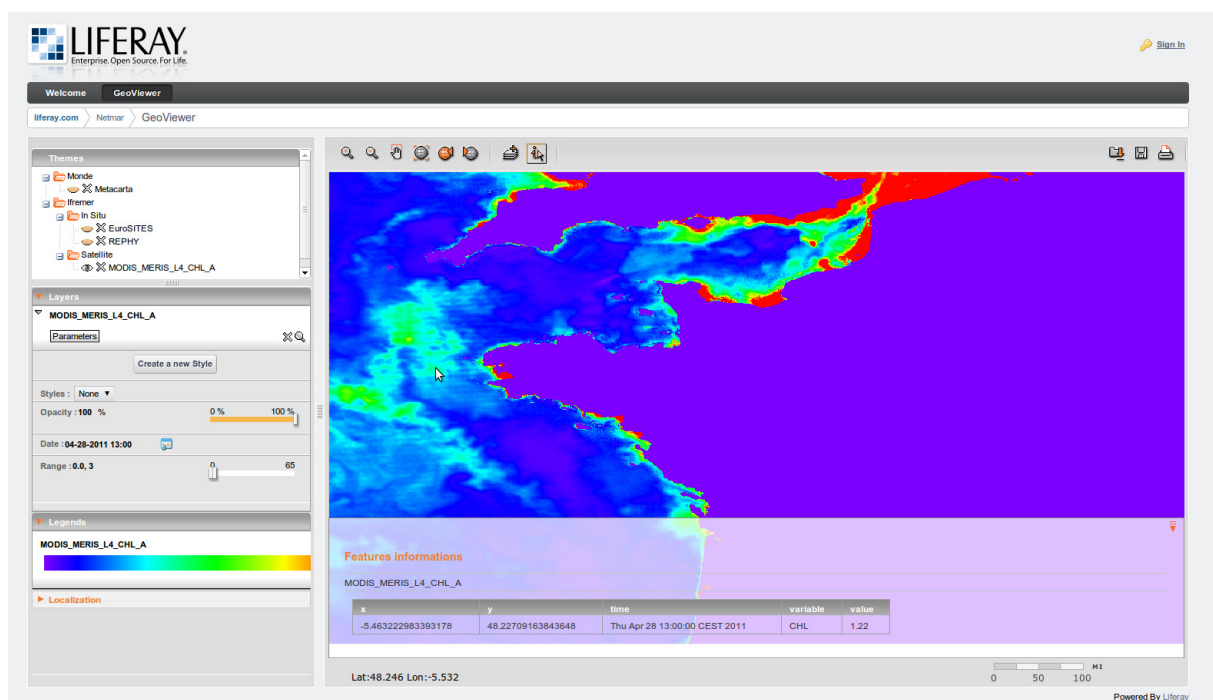


Figure 2-6 Example of satellite dataset (Chl-a from MODIS-MERIS) with some additional information: opacity, date selection, show colour manipulation, legend of the map (on the left panel: 'Layers'), and a feature request of chlorophyll on a point (below the map).

3 Smart search and discovery client

3.1 Objectives and features

Smart search is the ability of a discovery engine to locate target documents that do not include either an exact or fuzzy match to a user-supplied text string or user-supplied keyword. For example, if the user initiates a search on the discipline keyword 'Physical Oceanography' they find documents marked up with terms like 'currents', 'waves', 'water temperature', 'salinity', and so on.

The NETMAR smart search and discovery client aims to allow users to search data by meaning rather than by mere keywords. Version two of the smart search and discovery client supports three features: ontology browsing, dataset and service smart search, and metadata visualisation.

The smart search and discovery client will be used in three of the EUMIS pilots [PTS+11]:

- Pilot 1: Arctic Sea Ice and Met-ocean Observing System
- Pilot 2: Oil spill drift forecast and Shoreline Cleanup assessment services in France
- Pilot 4: The International Coastal Atlas Network (ICAN)

It provides search facilities based on the *semantics* of the data and services, including support for multi-domain and multi-lingual search in a network of metadata catalogue services connected by a mediator.

The second version of the smart search and discovery client has gone under major graphical re-design and bug fixes, and includes two additional capabilities: (i) support for multilingual ontologies and data and service search, and (ii) data access and visualisation component, that allows you to download available data and/or display them in the GIS viewer depending on the online delivery options.

3.2 Design and implementation issues

3.2.1 Design

The second version of the smart search and discovery client (semantic discovery client) includes three main components:

1. **Thesaurer**, which is the ontology browsing component;
2. **Geo Finder**, which is the smart data and service search interface (discovery GUI);
3. **Meta Viewer**, which is the metadata viewer.

The semantic discovery client relies on two web services: the semantic web service (SWS) and the CSW² mediator (CSWM), as illustrated in Figure 3-1.

² Catalogue Service for the Web (OGC specification)

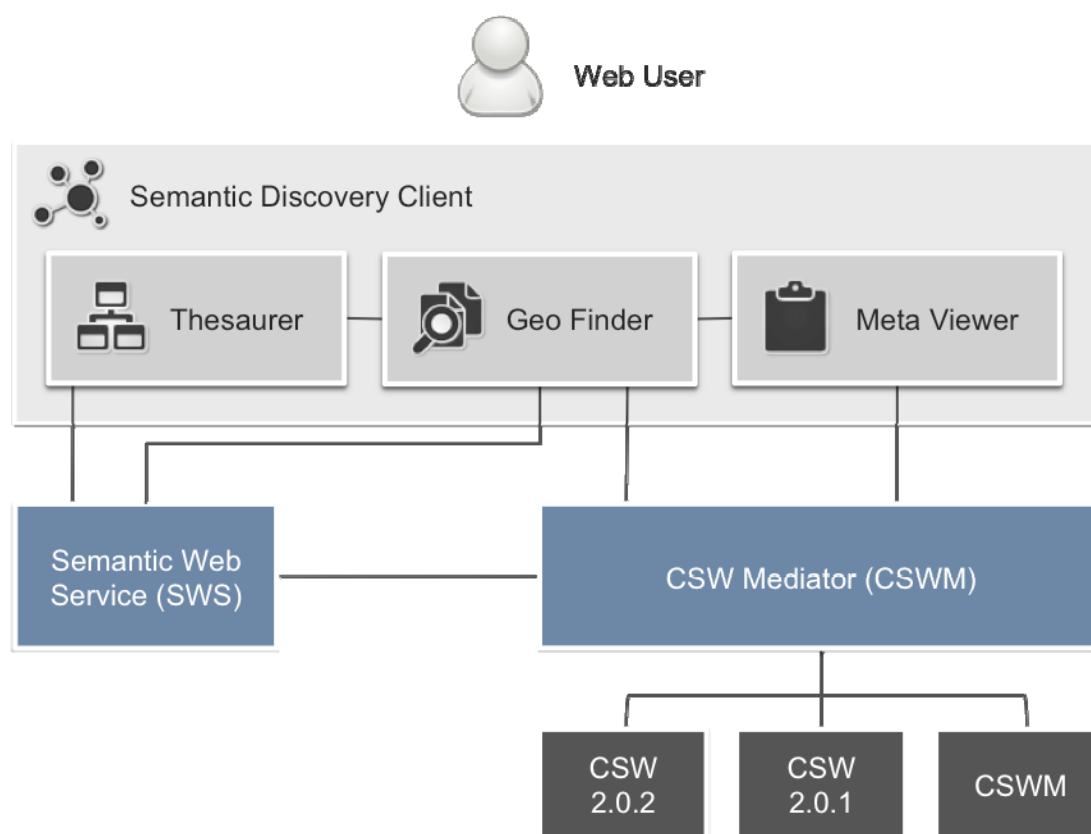







Figure 3-1 Semantic Discovery Components and Backend Services.

Metadata are delivered by distributed catalogue services that may be CSW 2.0.1, CSW 2.0.2, or CSWM. Currently, the EUMIS integrated 5 catalogue services, which are described in Table 3-1.

Table 3-1 NETMAR CSWM Catalogue Nodes

	NERSC catalogue service	Interface	CSW 2.0.2
		Software	GeoNetwork
		Host	NERSC
	met.no catalogue service	Interface	CSW 2.0.2
		Software	GeoNetwork
		Host	met.no
	PML catalogue service	Interface	CSW 2.0.2
		Software	GeoNetwork
		Host	PML
	IFREMER & CEDRE catalogue service	Interface	CSW 2.0.2
		Software	GeoNetwork
		Host	IFREMER
	ICAN catalogue service	Interface	CSWM 1.0
		Software	NETMAR CSWM
		Host	CMRC

The semantic web service (SWS) provides a high-level interface for retrieving knowledge from the NETMAR ontologies. The operations supported by the SWS are high-level and easy-to-use operations oriented towards querying SKOS (Simple Knowledge Organization System) thesauri [MB09]. They constitute the common operations required by most semantics-based components such as the CSW mediator, the ontology browser, and the discover GUI. Examples of such operations are: *Get Concept Schemes*, *Get Concept Hierarchy*, *Get Related Concepts*, and *Search Concept*. The SWS supports the KVP query types specified in the NETMAR report D4.3.1 and returns SKOS resources encoded in RDF/XML (Resource Description Framework/extensible Markup Language) [RDF04]. The SWS is developed in Java, using the Jena³ ontology framework.

On top of the catalogue services, a CSW mediator is responsible for handling metadata semantics and allows data discovery based on semantics. The CSWM supports distributed catalogue services, thus allowing new catalogue services to be integrated easily. Its main role is to:

1. Rewrite user requests by translating the keywords therein into equivalent and more specific (narrower) ontology concepts;
2. Submit the rewritten queries to the distributed catalogues;
3. Retrieve the responses and aggregate them before delivering them to the client application (e.g., the semantic discovery client).

The CSWM supports CSWM and CSW 2.0.2 XML requests and returns either a CSWM or a CSW 2.0.2 response (depending on the request standard). It is developed in Java

Thesaurer, the ontology browser, allows end-users to browse the ontologies with the aim of finding datasets or services of interest. Thesaurer allows users to better understand the meaning of the various topics (disciplines, themes, instruments, parameters, etc.) and how they relate to each other, and find in an accurate way data of relevance to their topics of interest. It allows them to search (by browsing or by keyword) concepts of interest, and from there navigate to the data discovery interface to find related datasets or services. Thesaurer supports multilingual ontologies, by allowing users to search terms in a one language and display results in a different language, if the languages in question are supported by the ontology.

The smart data discovery interface, called Geo Finder, allows end-users to search datasets of interest based on the “meaning of a keyword” and a geographic area of interest. Geo Finder includes an ontology browser for place keywords that hold geospatial information encoded in GML or JSON.

The metadata viewer, called Meta Viewer, is used to display metadata records of interest to the user. It allows users to download available data and/or display them in the GIS viewer depending on the online delivery options.

All of the smart discovery client sub-components (Thesaurer, Geo Finder, and Meta Viewer) are developed in Adobe Flex 3.5 and ActionScript 3.0.

3.2.2 Use Case 1 – Ontology Browsing

Figure 3-2 shows the sequence of interactions between the components involved in ontology browsing.

³ <http://jena.sourceforge.net/>

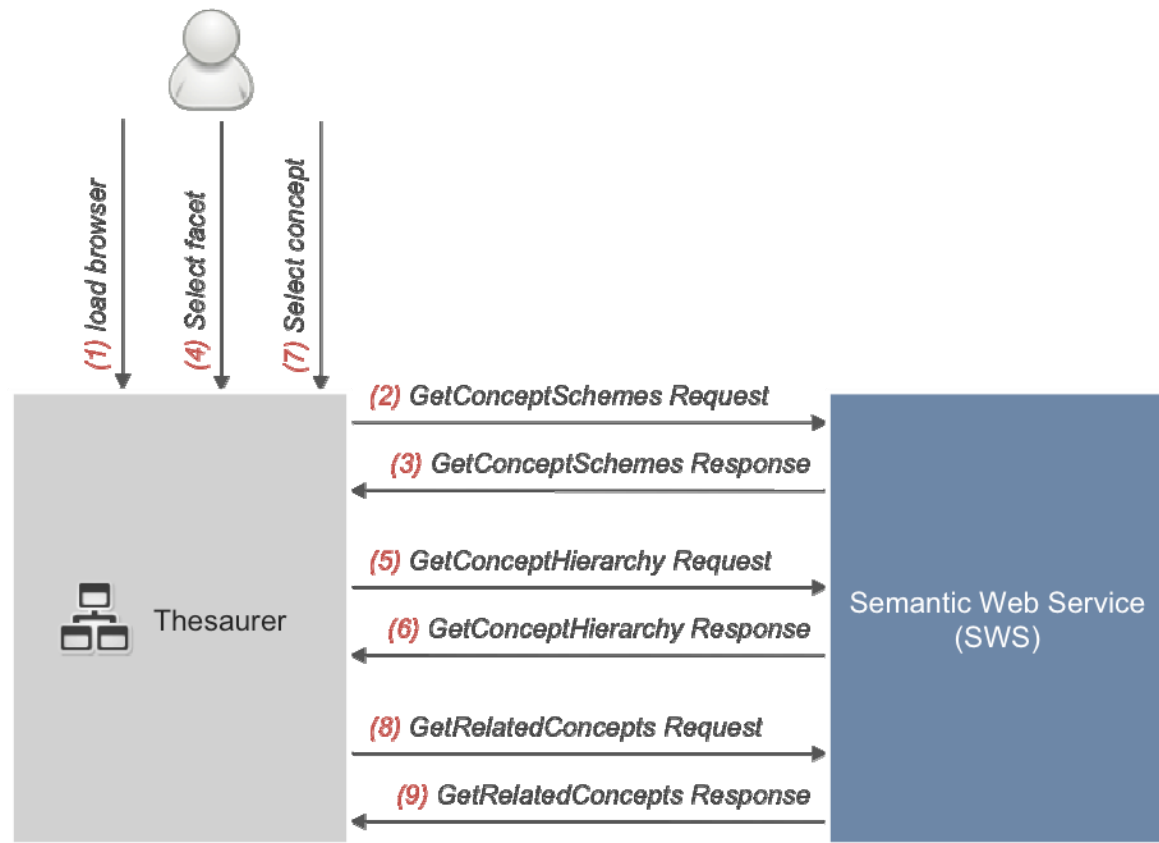


Figure 3-2 Protocol Diagram for ontology browsing in EUMIS.

As shown in Figure 3-2, only the ontology browser (Thesaurer) and the semantic web service (SWS) are involved in the ontology browsing use case. The typical component interactions illustrated in this figure are explained below.

- (1) The user loads the ontology browser.
- (2) While loading, the OB submits a *GetConceptSchemes* request to the SWS. The objective is to get all information (definitions, labels, etc.) related to the ontology browsing facets (different aspects of a categorisation of search terms). The ontology browsing facets (same as data discovery facets) agreed upon by all case studies are:
 - a. Instrument
 - b. Platform Class
 - c. Platform
 - d. Parameter
 - e. Vertical Coverage
 - f. Project
 - g. ICAN Coastal Erosion Themes
 - h. Oregon Coastal Atlas Themes
 - i. MIDA Themes
 - j. Service Taxonomy

Browsing facets are implemented as concept schemes in the NETMAR ontologies.

- (3) The SWS returns the list of concept schemes together with their definitions and labels which are used then by the browser to display the names (labels) and definitions of the facets.
- (4) The user selects the ontology browsing facet of interest.
- (5) The ontology browser sends a *GetConceptHierarchy* request to the SWS with the aim of retrieving the hierarchy of the concepts contained within the concept scheme (facet) selected by the user.

- (6) The SWS returns the concept hierarchy for the concept scheme selected by the user and the ontology browser displays it.
- (7) The user selects a concept of interest from the concept hierarchy.
- (8) In order to display the list of concepts related to the selected concept, the ontology browser sends a GetRelatedConcepts request to the SWS.
- (9) The SWS responds back and the ontology browser displays the related concepts.

The interactions can continue from step 4 (user selects a different concept scheme, or selects a related concept from a different concept schemes) or step 7 (user selects a related concept from the current concept scheme).

Following step 6 or 9, the user may select a concept from the ontology browser and request to search datasets or services related to the selected concept (c.f., next use case).

3.2.3 Use Case 2 – Data and Service Search

The interactions made between the EUMIS components during data and service discovery, which is illustrated in Figure 3-3, have been simplified in the second version of the semantic framework. Only one interaction (request-response) between the CSW mediator and the SWS is now required, as opposed to the sequence of three interactions (GetConcept, SearchConcept, GetRelatedConcepts) used in version 1. This has been achieved through the implementation of a convenience SWS method, called InterpretKeyword, which aggregates the previous three calls.

The components involved in this use case are: the semantic discovery client (GUI), the CSW mediator (CSWM), the semantic web service and the distributed catalogue services (catalogue nodes). A catalogue node may be a CSW (version 2.0.2 [CSW202] or 2.0.1 [CSW201]), or a CSWM.

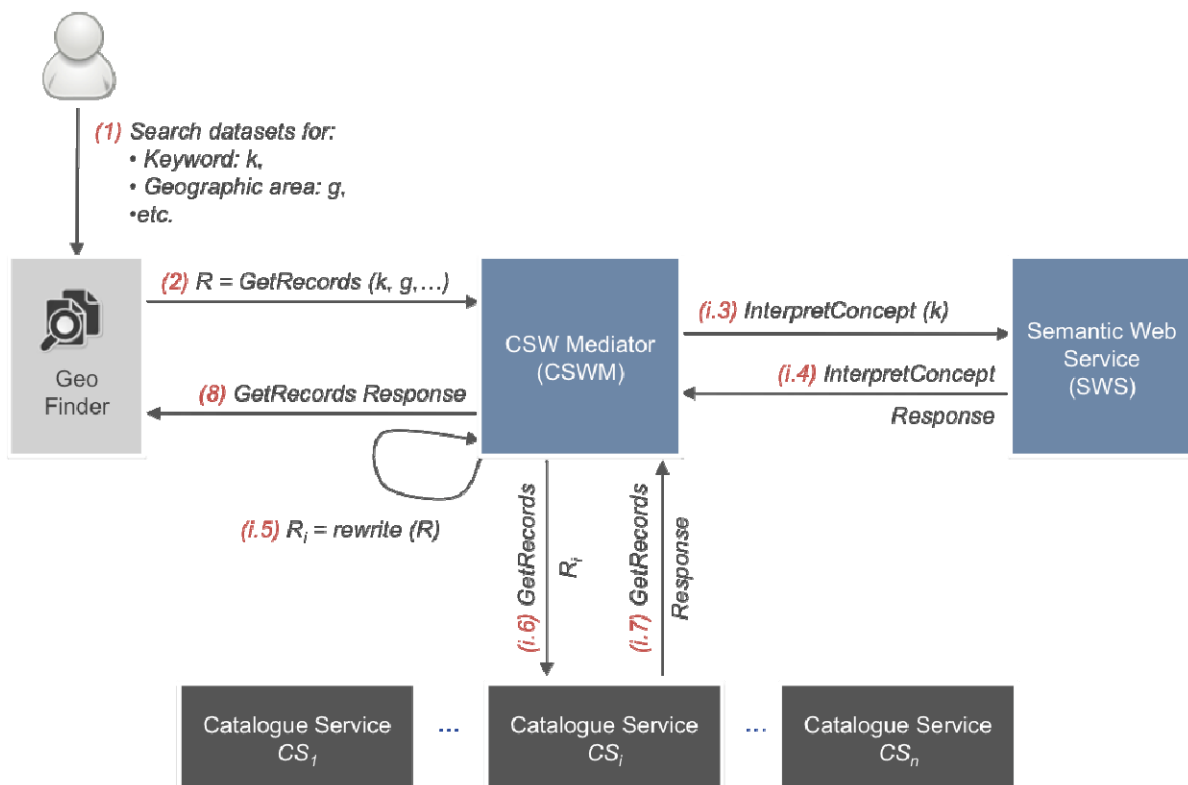


Figure 3-3 Protocol Diagram for data and service discovery in EUMIS.

The typical interactions illustrated in Figure 3-3 are explained below.

- (1) The user makes a request to the search component (Geo Finder) to search datasets (or services) for a given keyword k and a geographic extent g . Other search criteria may be included (e.g. temporal extent, vertical extent, etc.).
- (2) Geo Finder generates a CSWM GetRecords request R using the user's parameters (keyword k , geographic area g , etc.).
- The CSWM parses the user query R , and for each catalogue service CS_i to be involved in the query (by default all the catalogue nodes are involved unless otherwise specified in the user request R) it does the following (in parallel):
 - (i.3) The CSWM extracts the keyword of interest k and submits an InterpretKeyword request to the SWS with k as the keyword parameter value.
 - (i.4) The SWS responds to the CSW by sending the list of concepts semantically covered by the user keyword k (i.e., best matches of the keyword and their narrower concepts).
 - (i.5) The CSWM, now, has all the concepts covered by the user's keyword. It rewrites the user's request into a request supported by catalogue node CS_i and translates the original keyword using the concepts obtained in step (i.4). Let R_i denote the so-obtained query, which is a CSW or CSWM GetRecords query.
 - (i.6) The CSWM submits the rewritten query R_i to catalogue node CS_i .
 - (i.7) Catalogue node CS_i returns a response for the CSWM's request R_i .
- (8) The CSWM mediator collects all the catalogue node responses, wrap them in a CSWM GetRecords request and sends them back to Geo Finder, which in turn displays them to the user.

3.2.4 Implementation Issues

Below are the main technical and implementation issues encountered.

1. In the NETMAR report D4.2, "Semantic Framework Strategy," we recommended the use of the `gmx:Anchor` element as a substitute for `gco:CharacterString` in metadata keywords. The objective is to include references to ontology concept URIs while keeping human-readable labels in keywords. However GeoNetwork does not support `gmx:Anchor`. As a work around, we encoded keywords as concept URIs inside the `gco:CharacterString` elements.
2. There is currently no standard for modelling geospatial concepts (e.g., place keywords). In our approach, we define geographies of concepts as JSON text embedded in the concept definitions.

3.3 Examples of use

A screenshot of the discovery browser is shown in Figure 3-4. On the left side, a vertical navigation menu (toggle button bar) allows you to select the component of interest: Thesaurer, Geo Finder, or Meta Viewer.

In Figure 3-4, the selected component is the ontology browser (Thesaurer). On the left side of the ontology browser, a tree is used to display the topic hierarchy of a given facet (here the Instrument facet). When the user selects a topic (e.g., CTD), the ontology browser displays its definition and related topics (bottom right). By clicking on a related topic, the ontology browser navigates to it in the topic hierarchy.

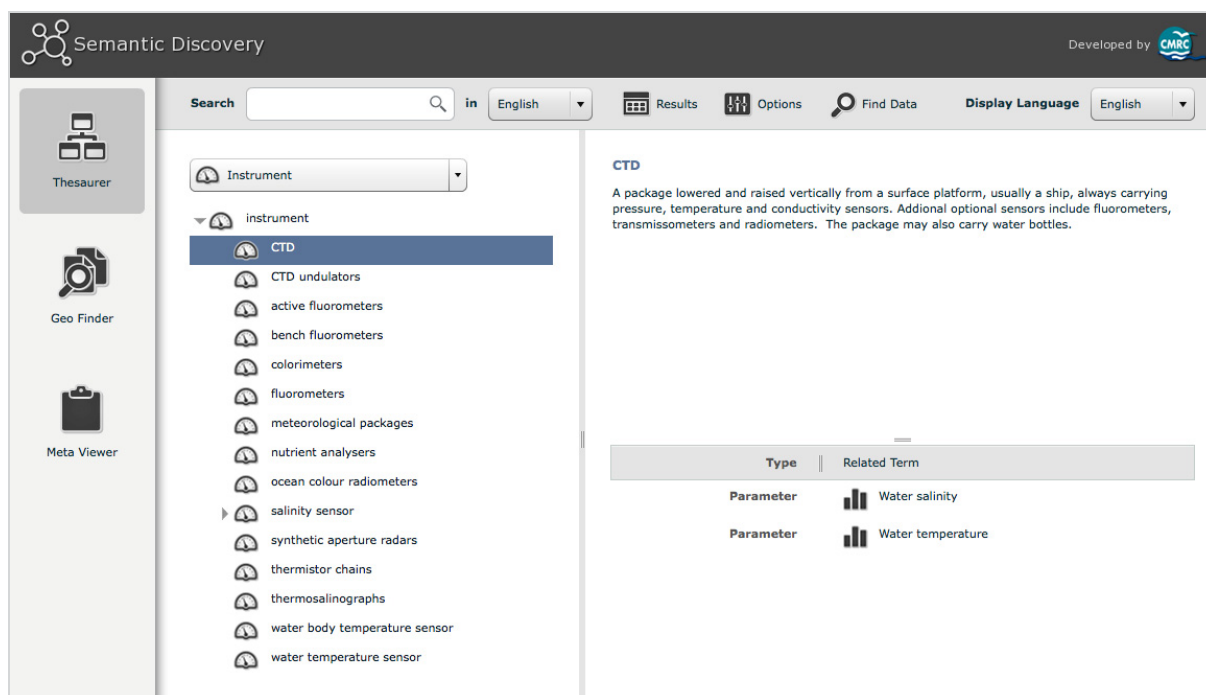


Figure 3-4 Screenshot of the Semantic Discovery Client Showing the Ontology Browser (Thesaurer).

Users can search topics by keyword. They can specify the keyword language (e.g., French), the display language (i.e., result language, e.g., English), and the target search facets. The screenshot in Figure 3-5 shows a use case where the user is searching for the French term “*activité humaine*”, and requests that results be displayed in English (display language).

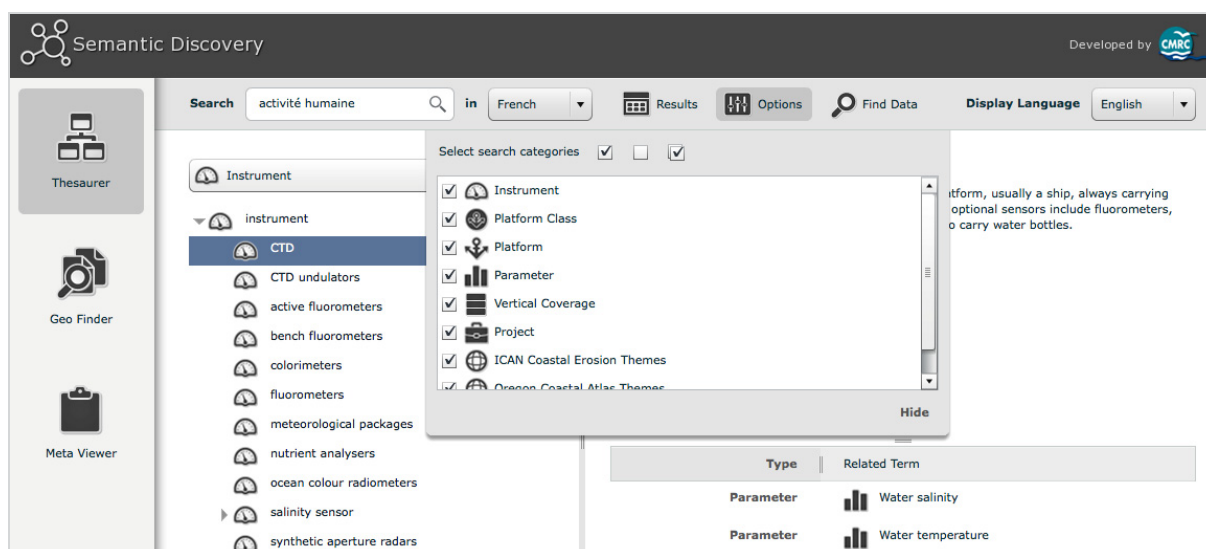


Figure 3-5 Concept Search Options in the Ontology Browser.

The next figure (Figure 3-6) shows a screenshot of the OB search results for the French term “*activité humaine*”.

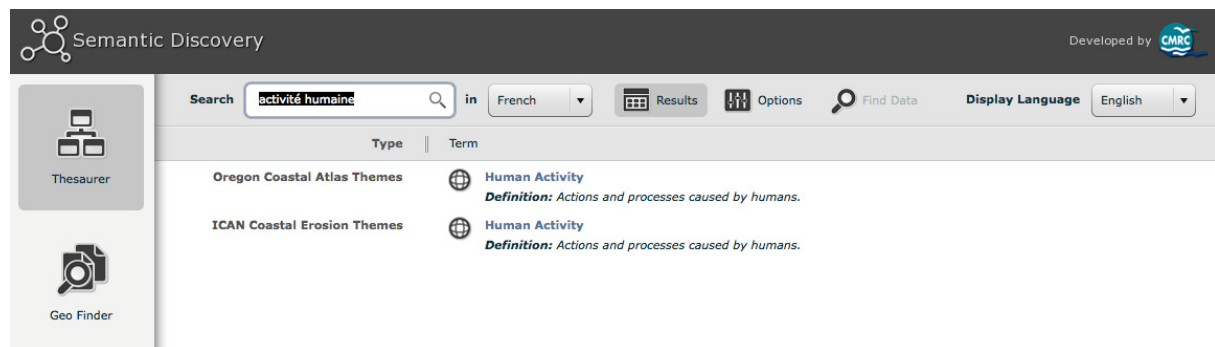


Figure 3-6 Topic Search in the OB.

When a user clicks on a result (topic), the ontology browser navigates to it in the topic hierarchy and displays its synonyms, definition, and related topics as per Figure 3-5.

By selecting a topic in the topic hierarchy, and clicking on the “Find Data button” (see Figure 3-5) the semantic discovery client navigates to the Geo Finder tab and initialises it with the corresponding keyword.

Figure 3-7 shows the data search interface. In this screenshot, the keyword had been initialised by the ontology browser to the URI of the concept “Coverage (by area) of ice on the water body by Advanced Microwave Scanning Radiometer for EOS (AMSRE-E)”. But free text keywords may be entered manually as well. A geospatial ontology browser (left side) allows users to select a place from the NETMAR place ontology and display its bounding box in the map area (in Figure 3-7 the selected area of interest is the North Sea). Alternatively, a user can draw a bounding box.

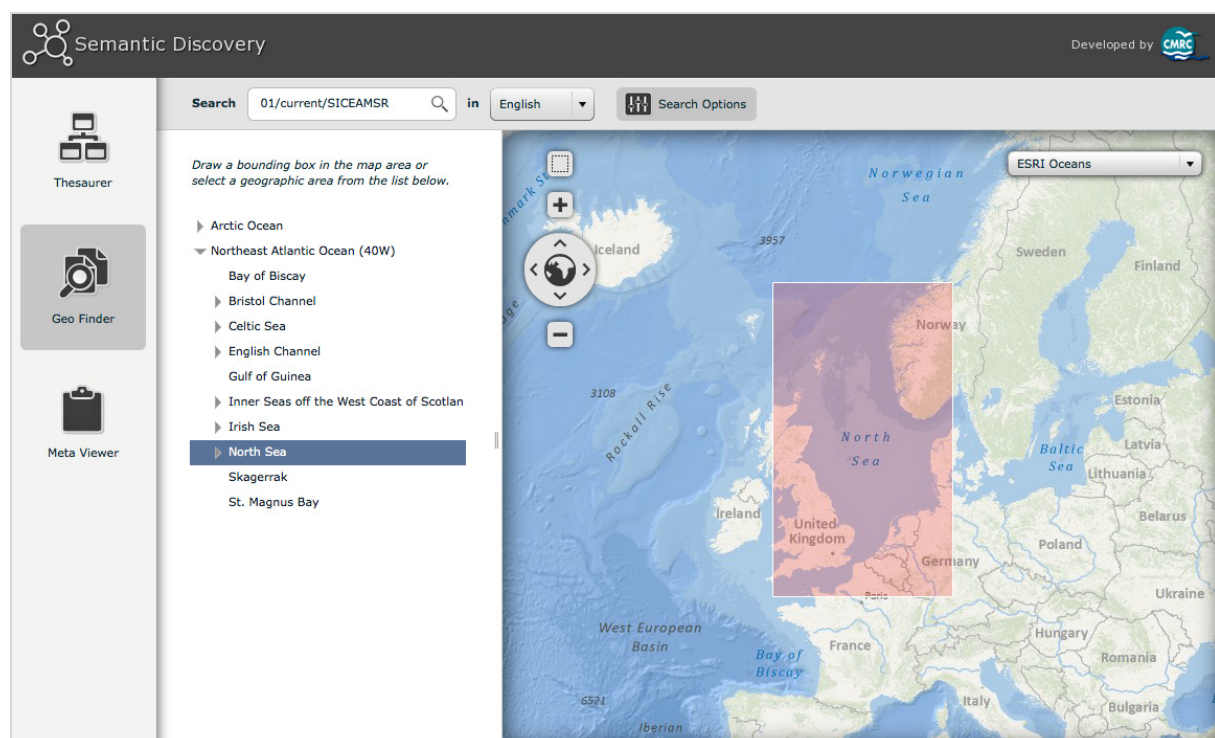


Figure 3-7 Data Discovery Interface.

By pressing the search icon inside the search text input, or by pressing Enter, the ontology displays the results, and the user can select a result in order to display it in the metadata viewer (c.f., Figure 3-8). Like in the ontology browser, the keyword language is required for

searching data (default is English). Depending on the search criteria (keyword and area of interest), results may come from one or many catalogue services. The example in Figure 3-8 shows only one record, which is returned by the met.no catalogue service.

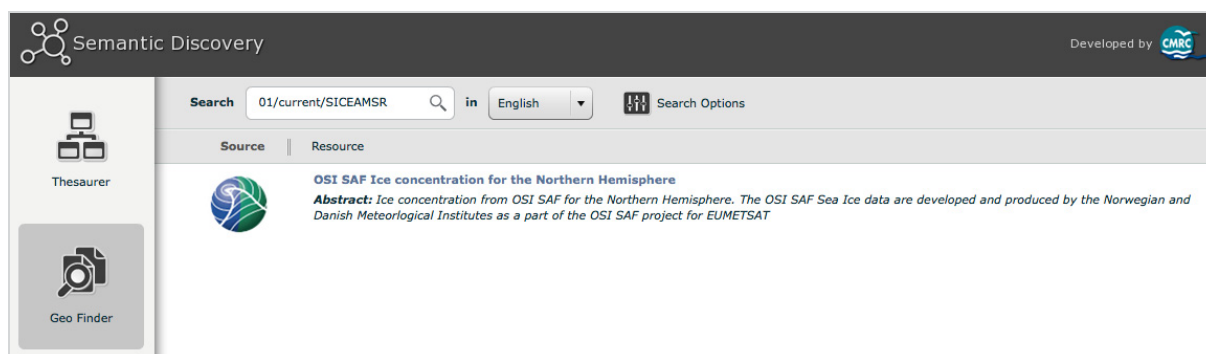


Figure 3-8 Data Discovery Interface.

By clicking on a search result, the discovery client navigates to the metadata viewer (Meta Viewer), and opens the corresponding metadata record that is fully loaded from the source catalogue service through the CSW mediator. The Meta Viewer interface (c.f., Figure 3-9) offers four views (tabs): summary metadata (Summary), full metadata (Full), XML (XML), and data distribution information (Data Links).

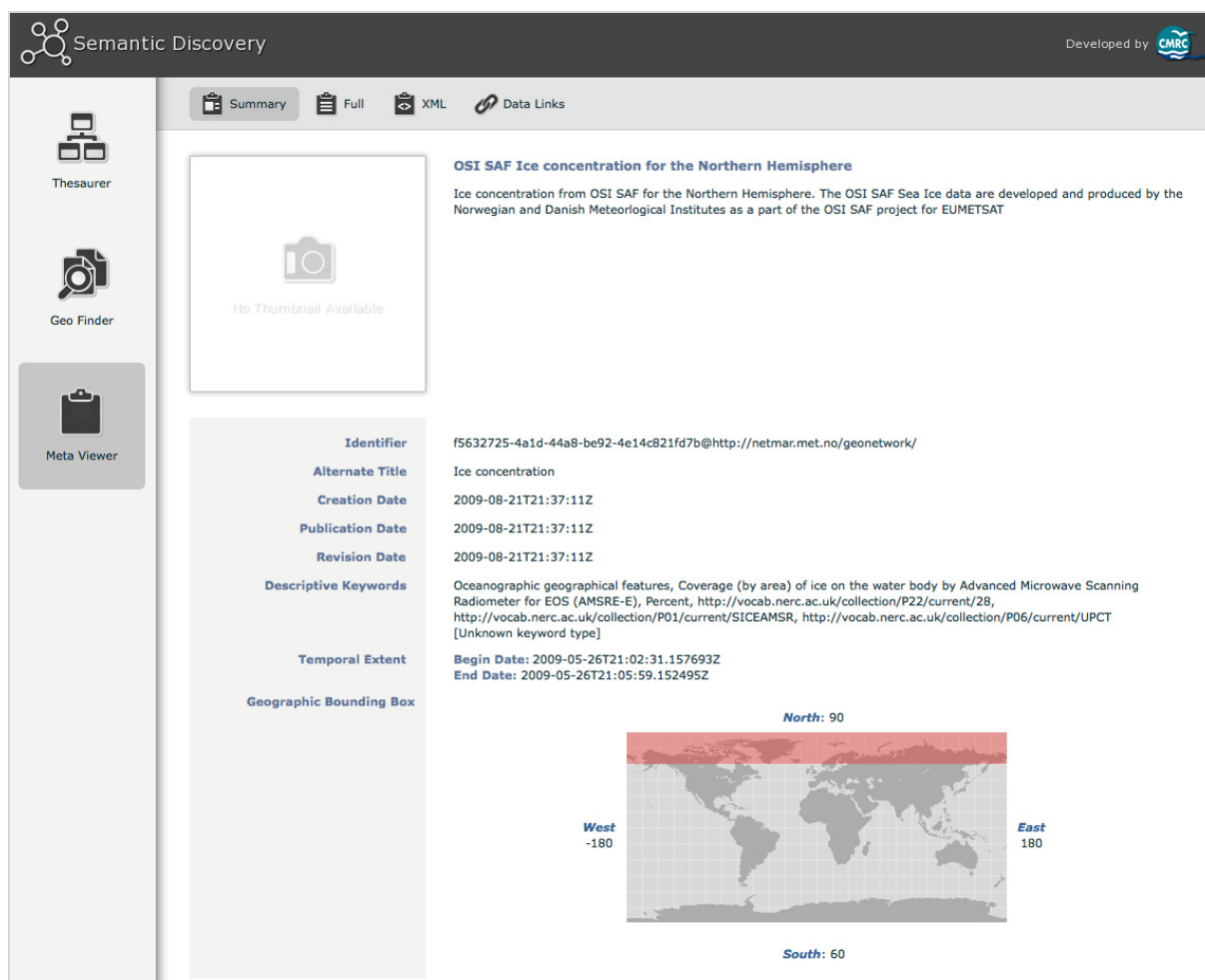


Figure 3-9 Data Discovery Interface - Summary Metadata.

Figure 3-10 shows a screenshot of the metadata viewer's data links tab, which contains the list of the online data distribution options (left side), and the details of each option (right side). Depending on the distribution option (e.g., WMS, WFS, file download) the user may perform different actions, e.g., download static map, display in EUMIS GIS map viewer (section 2), download GML data from WFS, download data file, etc.

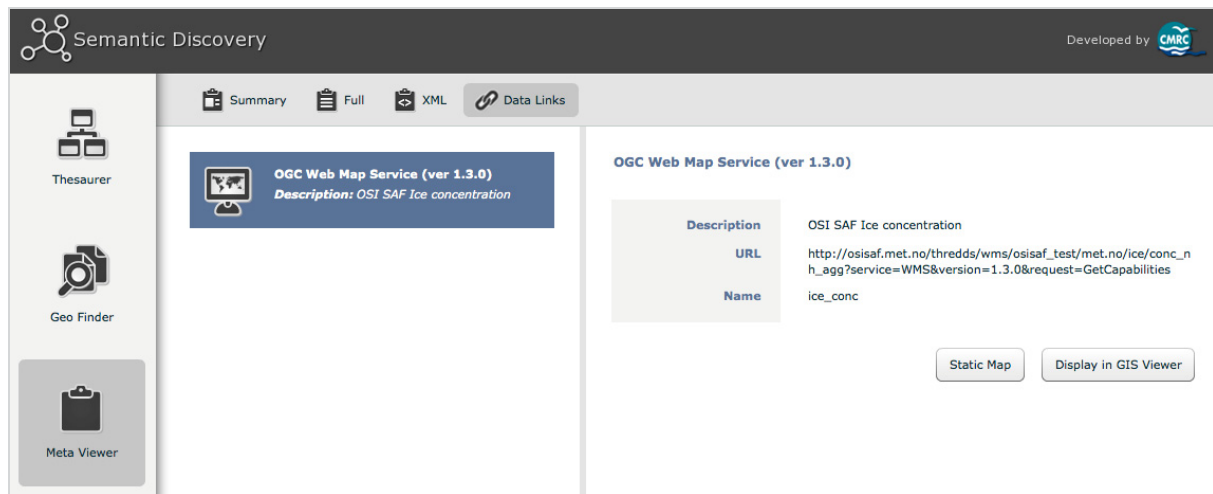


Figure 3-10 Data Discovery Interface - Data Delivery Options.

4 Service chaining editor

4.1 Objectives and features

One key aim of the NETMAR project is to enable non-technical users to build their own workflows using components (data and processing) provided by the project. These workflows could then be saved and reused by the user or published for use by others. A user requirement identified already during the proposal stage was that the editor must be available online via any commonly used web browser, as part of the EUMIS portal. The alternative of requiring the user to download specialist software to create and edit service chains, then upload the chain description, is an impediment to ease of use. Furthermore, this online editor will be available under an open source license to other projects and will be a significant contribution to simplifying workflow design.

Taverna, the workflow engine selected for NETMAR, provides a Java workbench which can be run locally to develop workflows [Tav]. Other systems such as Kepler [Kep] and various BPEL (Business Process Execution Language) [O07] based programs also provide this client side facility.

In deciding on the functionality to be provided within the NETMAR service chaining editor, we have taken advantage of the extensive user interface testing by many scientific users of the current Taverna workbench, and drew many of the initial requirements from there.

These desktop workflow editors have a number of features in common:

- Drag and Drop properties
- Save/Load workflow
- Validation of workflow
- Service description based on a web protocol (e.g. WSDL (Web Services Description Language) [W3C07], or RESTful [LR07])
- Connection between services that compose workflow
- Output-to-Input connection (output terminal/port shall only connect to input terminal/port)
- Generation of workflow description to run in engine (e.g. Apache ODE [AO], Taverna)
- Saving/Loading previous states

The NETMAR service chaining editor provides these capabilities within the web interface and the final version will also include facilities to make use of semantic information to help assemble a valid workflow. The service chaining editor will be used to develop customised workflows for the users of the EUMIS pilots [PTS+11]:

- Pilot 1: Arctic Sea Ice and Met-ocean Observing System
- Pilot 3: Ocean Colour – Marine Ecosystem, Research and Monitoring.

4.2 Design and implementation issues

4.2.1 Client side editor GUI

The current service chain editor supports the following functionalities (Figure 4-1):

- WSDL support (via URL). The user may load services from any compliant server.
- Workflows may be saved within local browser storage (Workspace) for future editing.
- Workflows may also be downloaded/uploaded as JSON objects from the client machine's file system.
- Workflow export in two formats: NETMAR XML and Taverna 2.3 [Tav]. These workflows may then be uploaded to the orchestration server to be executed.

- Workflow creation using drag and drop of processes defined in WSDL.
- Workflow zoom control, allows larger and more complex workflows to be edited.

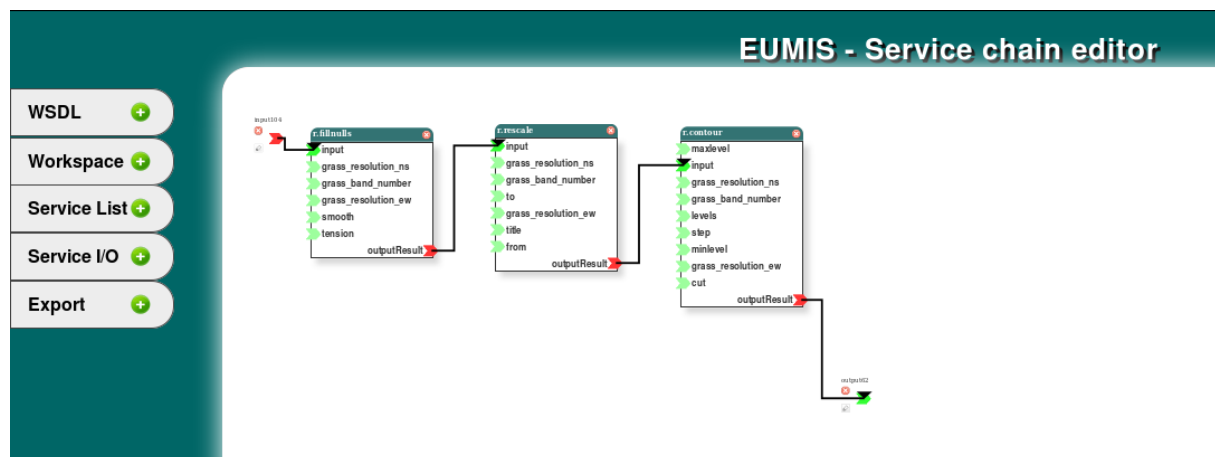


Figure 4-1 Generic example of service chain editor.

4.2.2 WSDL support

NETMAR processing services are built to allow access through SOAP (Simple Object Access Protocol) and provide information on individual process capabilities in a WSDL document. This WSDL document is then used by the editor to determine the availability of processes and how they may be chained. The process is summarised below.

- A WSDL file is referenced by the user by introducing an URL with its location
- The editor parses the WSDL file describing the services, and assembles a set of processes, each represented as a box with specific I/O based on the description.
- WSDL parsing is done within the client
- WSDL is fetched using a PHP proxy installed in the server to avoid cross-site XML issues

Web services follow a SOA strategy where code will have to call services or access resources located in other remote servers. JavaScript prevents code in the user's browser accessing any service that is not located on the host providing the web page [YDN]. This problem is overcome by the use of a proxy service on the host providing the service editor; this proxy will fetch content/services required by the service editor. The current version uses a PHP proxy to fetch WSDL XML content and will check for XML validity and prevents any content other than XML being returned to the editor. This WSDL XML file, containing the definition of a set of WPS services, is then transformed to a JSON object, which is used by the service chaining editor to construct nodes for each WPS service, so that these can be graphically manipulated in the WYSIWYG editor and combined to make up a workflow. The WSDL definition is also used when serialising the constructed workflow so that a workflow engine later can call the chosen WPS services with the correct input parameters and obtain the results according to the defined output parameters.

4.2.3 Workspace, see Figure 4-2

Current technologies like HTML5/jStorage [JS] allow for local data storage without proprietary plugin installation. The workspace component of the service chaining editor allows the user to store created workflows within the browser instance and, if desired, to restore previously saved versions. The workspace upload feature enables workflows to be transferred to the user's computer as JSON-objects.

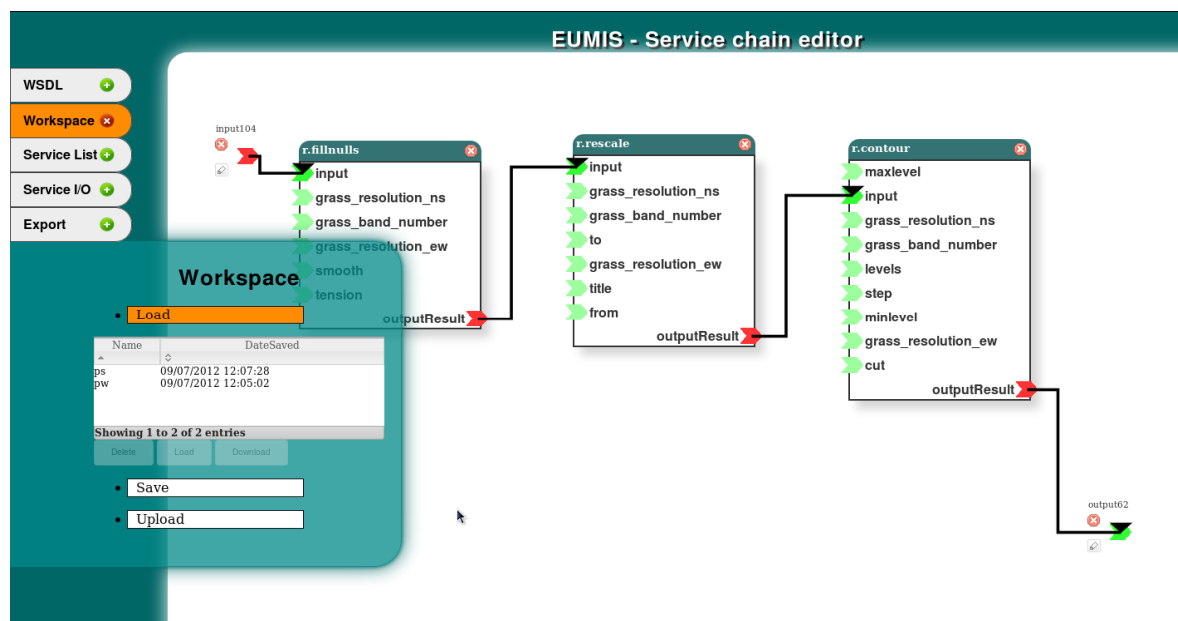


Figure 4-2 Example of workspace (left side)

4.2.4 Workflow export, see Figure 4-3

The service chain editor uses a JSON-object to describe the workflow's structure. This object can be converted into XML formats that are easier to integrate in third party software. Two formats are currently defined:

1. The NETMAR XML is a simple XML description of the JSON object using the JSON2XML API [JX].
2. T2flow file support for Taverna version 2.3 is provided by a PHP service that runs a XSLT transformation based on the NETMAR XML, returning a fully functional T2flow XML file that can be used in Taverna's workbench and server software or uploaded to the NETMAR orchestration engine.

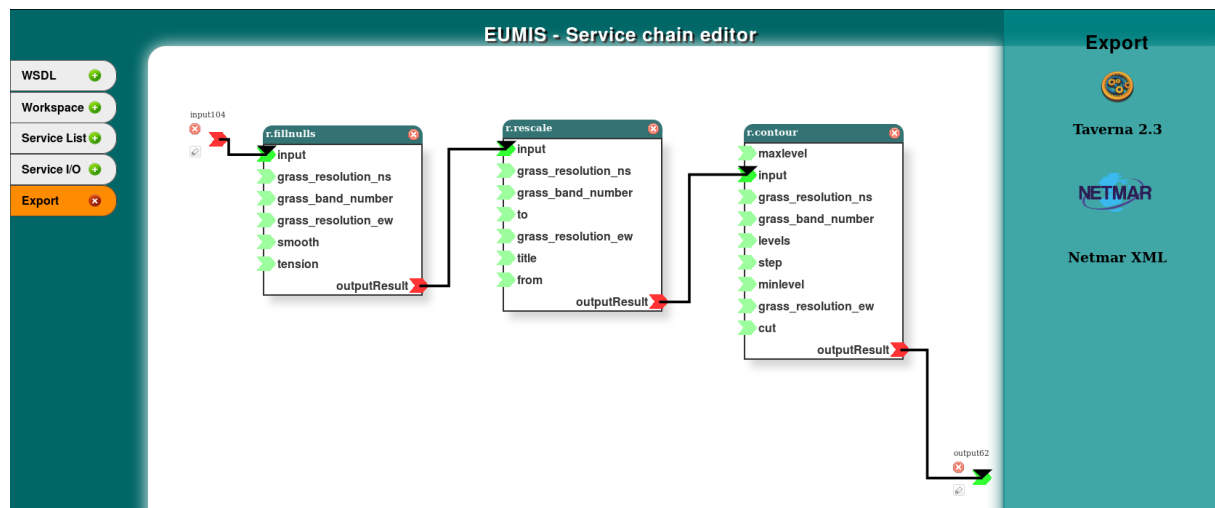


Figure 4-3 Export functionalities into NETMAR XML format or Taverna 2.3 (T2flow).

4.2.5 Zoom control

Workflows will contain many processing services, represented as boxes. Each box may occupy a considerable space, especially if its list of Input/Output terminals is big. By using the features of CSS 2D [W3CCSS], it is possible to zoom in/out of the workflow structure in

order to edit larger workflows (Figure 4-4). The current version allows 4 levels of zoom (100%,75%,50%,25%).

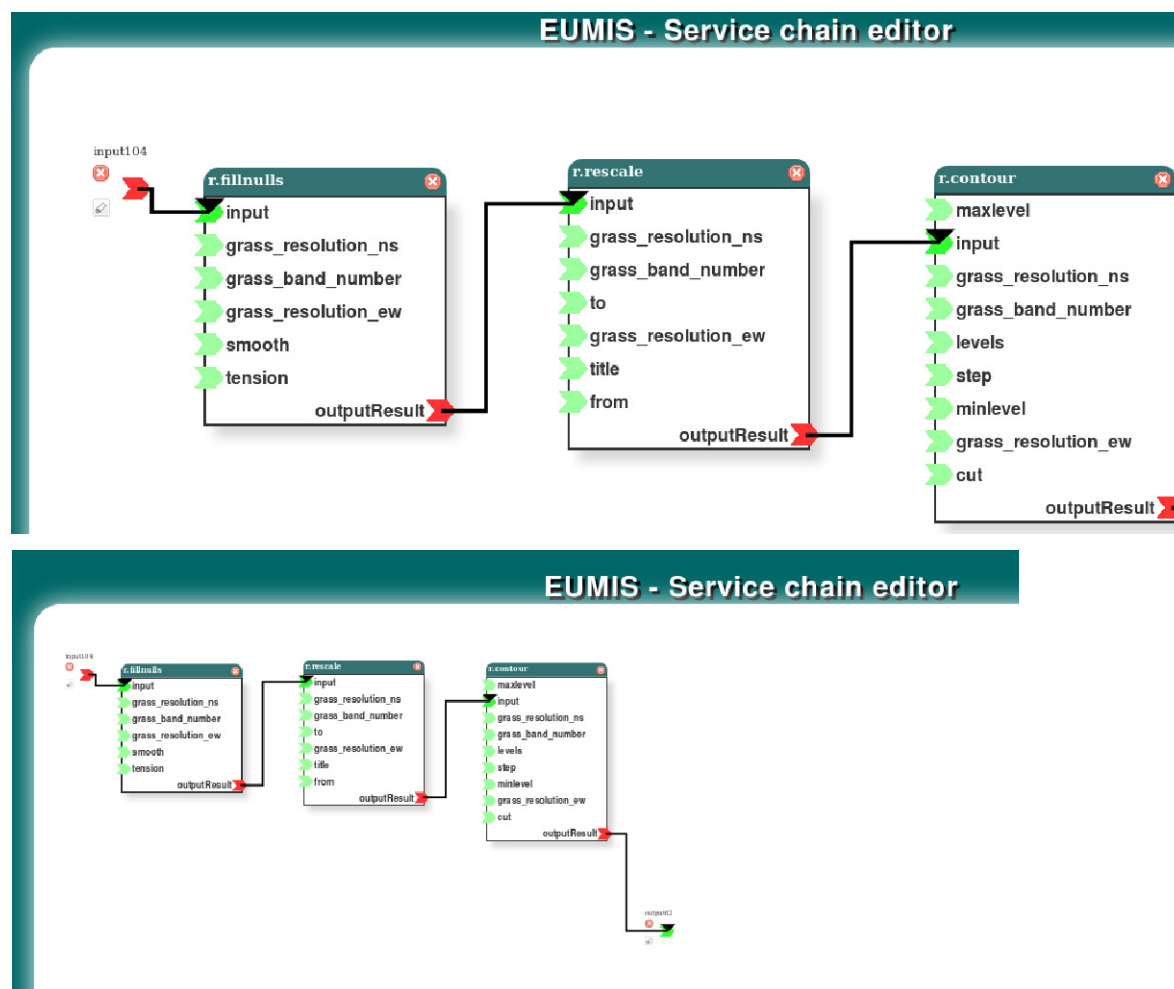


Figure 4-4 Example of zoom, same workflow at 100% and 50% zoom.

4.3 Usage examples

4.3.1 Combining Satellite and Model Data

A typical use case would be that of a scientific user who has a dataset containing, say, chlorophyll values at a particular resolution. If this dataset is to be compared with a model hindcast then:

- It must first be converted to the same resolution as the model.
- If the areas covered by the model and the dataset differ then they should have the intersecting area extracted to make the comparison easier.
- The 2 extracted images will then be passed to a comparison process which will carry out the statistical analysis as specified by the scientist.

This would be achieved in the service chaining editor by opening the standard tools selection and choosing the resample tool, area extractor and statistics tool and connecting them together. See Figure 4-7 Workflow example for model and satellite comparison.

4.3.2 Simple WebGIS facilities

Another example shown in Figure 4-5 illustrates a raster input being converted to a contour map for analysis. This would normally be carried out within a client based GIS but in this case could be built as a web service within EUMIS.

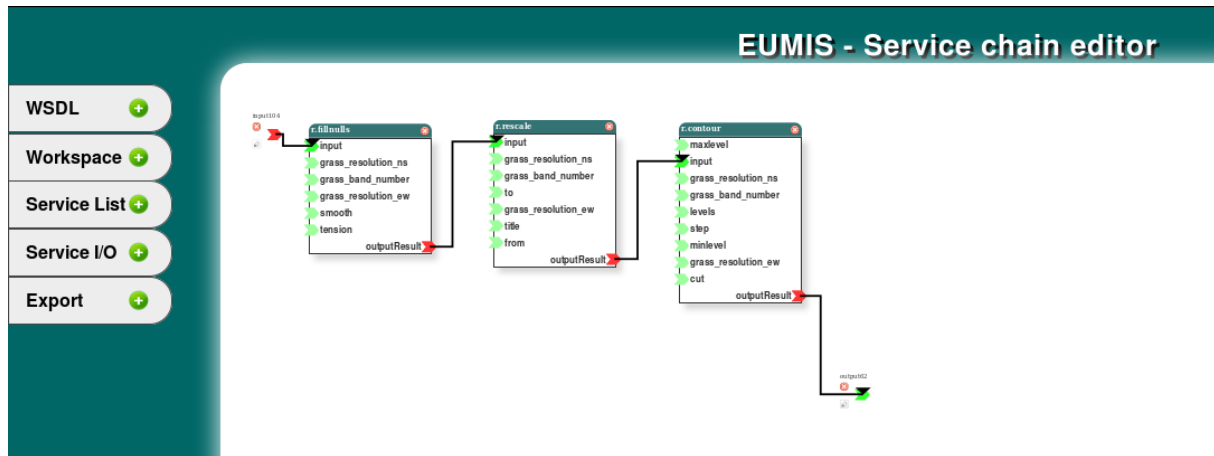


Figure 4-5 Example workflow using *r.fillnulls*, *r.rescale* and *r.contour*. Web services provided by GRASS-GIS and PYWPS4 5.

Here a chain has been built to process a raster input, filling any null values in the raster by means of interpolation (*r.fillnulls*), rescaling the image (*r.rescale*) and passing the output to a contouring process (*r.contour*) to generate a vector output that can be used more simply for analysis purposes. Each node in the workflow originates from the WDSL file definition, which is read into the editor through the WDSL option in the left hand menu.

4.3.3 Workflow execution

The service chaining editor will translate the user's service choice and chaining relations into a T2Flow/SCUFL2 XML format [SCUFL2]. The generated XML can be sent to the Taverna engine which will orchestrate the service chain (manage data flow between services, check service availability, deal with exceptions and errors raised by services and finally send the final output of the workflow back to the portal) (Figure 4-6). The nodes in the service chain (i.e. workflow) each represents a WPS service, which is called by the workflow engine with the appropriate input according to the service definition.

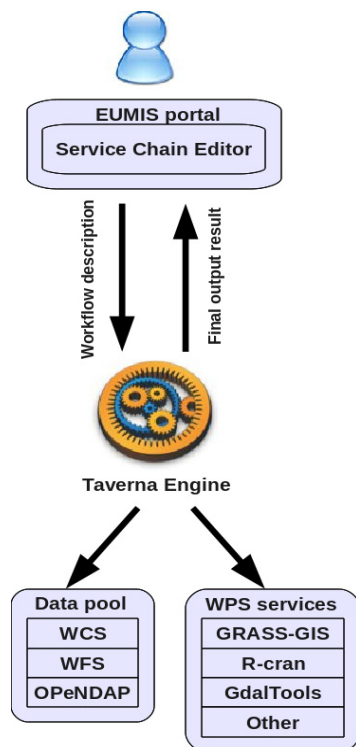


Figure 4-6 Service orchestration strategy.

For example, when combining satellite and model data the Taverna engine would orchestrate a workflow that would transform the data, as described in section 4.3.1; the data would be fetched from WCS based on time and area, its resolution changed, non overlapping areas clipped, statistical analysis run, and the calculated output would be transferred back to the portal. This is illustrated in Figure 4-7.

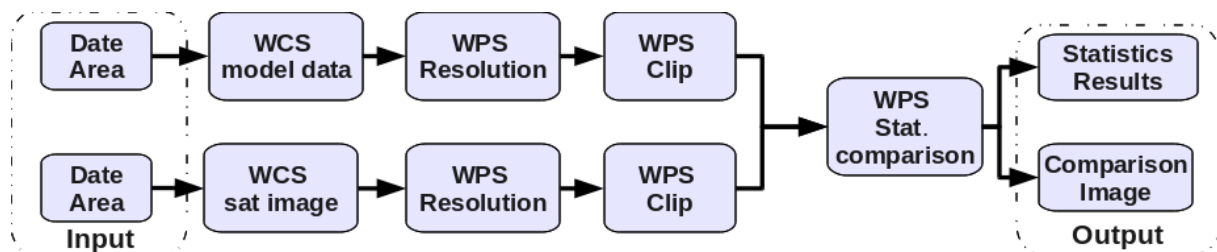


Figure 4-7 Workflow example for model and satellite comparison.

5 Conclusions

The GIS map viewer offers common GIS operations, such as layer selection and display, simple colour manipulation of raster layers, zooming and panning, and restoring the displayed map to the original extent. It can connect to multiple WMS servers, and these can be configured dynamically by the user, i.e. which WMS server to retrieve data from can be set and changed at run time. The colour scale used to display raster maps can also be manipulated, allowing the user to select the colours that gives the best contrast on the screen. For in situ stations, the location is shown by a symbol annotated with the station number. If the WMS server and layer support it, the user can get additional information, such as the observational values, by clicking on the station symbol. The GIS map viewer supports multiple map projections and also allows execution of a remote WPS service. The GIS Map Viewer was implemented as a portlet and can be integrated directly in the EUMIS portal.

The smart search and discovery client allows the user to browse the hierarchy of concepts in the NETMAR ontology, navigate within and across multilingual and multi-domain ontologies, and use the terms therein as a basis for data and services search. The user can also enter a keyword and (optionally) a geographic area as search criteria. The client queries a CSW (Catalogue Services for the Web) Mediator (CSWM) for products and services that match the given criteria. The CSWM is connected to a network of CSW metadata catalogues (nodes) that use the NETMAR ontologies to mark up their entries, and will query each of these and merge the individual results into one joint result. The CSWM can also query another CSWM, which in turn will query its network of CSW metadata catalogues and send back a joint result, which can be merged with results from other metadata nodes. The client is implemented as a Flex application, which must be wrapped in a JSP (Java Server Pages) or HTML (HyperText Markup Language) page before inclusion in the EUMIS portal.

The service chaining editor is offering a WYSIWYG (What You See Is What You Get) interface to making a workflow composed of a series of individual data and/or processing services. The services are 'chained' together so that the output from one or more services is provided as input to one or more other services. The whole service chain can be stored for later enhancement, and when converted to the Business Process Execution Language (BPEL) format, be executed by a workflow engine like Taverna. The service chaining editor is developed using JavaScript only, which must be wrapped in a portlet using JSP and HTML before being integrated in the EUMIS portal.

The three components in this report are ready for integration in the EUMIS portal. The GIS map viewer can be integrated directly, while the other two components must be wrapped in short snippets of JSP or HTML code and their interfaces described in portlet configuration files, before they can be integrated in EUMIS. The fully integrated components will be used in the final user demonstration of NETMAR, to be held in autumn 2012. Experiences from the integration and feedback from the users on the components will be reported on in later NETMAR Deliverables.

6 References

- [AO] Apache ODE, <http://ode.apache.org/> (accessed 11 June 2011)
- [CSDI] Constellation-SDI, <http://constellation.codehaus.org/> (accessed 11 June 2011)
- [CSW201] OpenGIS Catalogue Service Specification, Version 2.0.1, OGC 04-021r3. Online at http://portal.opengeospatial.org/files/?artifact_id=5929&version=2 (accessed 30 May 2011)
- [CSW202] OpenGIS Catalogue Service Specification, Version 2.0.2, OGC 07-006r1. Online at http://portal.opengeospatial.org/files/?artifact_id=20555 (accessed on 22 April 2010).
- [GS] Geoserver, <http://geoserver.org/display/GEOS/Welcome> (accessed 11 June 2011)
- [Fla11] Flanagan, Davidm, 2011. JavaScript: The Definitive Guide. O'Reilly Media; 6th edition (May 10, 2011).
- [ICAN] ICAN web site, <http://ican.science.oregonstate.edu/> (accessed 28 July 2011)
- [JQ] jQuery, <http://jquery.com/> (accessed 11 June 2011)
- [JS] jStorage, www.jstorage.info (accessed 10 Jul 2012)
- [JX] Json2XML, <http://goessner.net/> (accessed 10 Jul 2012)
- [Kep] The Kepler Project, <https://kepler-project.org/> (accessed 11 June 2011)
- [LL11] Lowry, Roy and Leadbetter, Adam, 2011. Prototype interlinked ontology resource. NETMAR (Open service for marine environmental data) Deliverable 3.6. European Commission Information Society and Media Directorate-General Grant Agreement Number 24902.
- [LR07] Richardson, Leonard and Sam Ruby, 2007. RESTful Web Services Web services for the real world. O'Reilly Media.
- [LRv6] Liferay Community Edition, <http://www.liferay.com/downloads/liferay-portal/available-releases> (accessed 30 May 2011)
- [MF] MapFaces, <http://mapfaces.codehaus.org> (accessed 31 May 2011)
- [MS] MapServer, <http://mapserver.org/> (accessed 11 June 2011)
- [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/>
- [NCW12] ncWMS, <http://www.resc.rdg.ac.uk/trac/ncWMS/> (accessed 31 July 2012)
- [O07] OASIS, 2007. Web Services Business Process Execution Language Version 2.0. Online at <http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.pdf> (accessed 11 June 2011)
- [OD] OrchestraDesigner, <http://orchestradesigner.riaforge.org/> (accessed 30 May 2011)
- [OL] OpenLayers, <http://openlayers.org> (accessed 30 May 2011)
- [Oryx] Oryx editor, <http://bpt.hpi.uni-potsdam.de/Oryx/> (accessed 30 May 2011)
- [PTS+11] Pladsen, Jan Ivar, Mickael Treguer, Stein Sandven, et. al, 2011. D1.1 Use case specification and schedule for testing. NETMAR (Open service network for marine environmental data) Deliverable D1.1. European Commission Information Society and Media Directorate-General Grant Agreement Number 249024. Revised version, June 2011.

- [RDF04] RDF Working Group: Resource Description Framework, World Wide Web Consortium (W3C), 10 February 2004. Available at: <http://www.w3.org/RDF/> (accessed 11 June 2011).
- [RFC2396] Network Working Group, 1998. Uniform Resource Identifiers (URI): Generic Syntax, <http://www.ietf.org/rfc/rfc2396.txt> (accessed 11 June 2011).
- [SCUFL2] <http://www.mygrid.org.uk/dev/wiki/display/developer/2010-07+SCUFL2> (accessed 07 June 2011)
- [Tav] Taverna workflow management system, <http://www.taverna.org.uk/> (accessed 30 May 2011)
- [TDS12] Thredds Data Server, <http://www.unidata.ucar.edu/projects/THREDDS/tech/TDS.html> (accessed 31 July 2012)
- [YDN] Yahoo! Developer Network, "JavaScript: Use a Web Proxy for Cross-Domain XMLHttpRequest Calls", <http://developer.yahoo.com/javascript/howto-proxy.html> (accessed 30 May 2011)
- [W3C07] W3C, 2007. Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language. Online at <http://www.w3.org/TR/wsdl20/> (accessed 11 June 2011)
- [W3CCSS] W3C, 2012. CSS Transforms. W3C working Draft 3 April 2012 (accessed 10 Jul 2012)
- [WI] WireIt!, <http://neyric.github.com/wireit/> (accessed 30 May 2011)
- [WAPI] WireIt! API, <http://neyric.github.com/wireit/api/> (accessed 30 May 2011)
- [WMS13] OGC Web Map Service (WMS) Implementation Specification, version 1.3.0, OGC 06-042, Online at http://portal.opengeospatial.org/files/?artifact_id=14416 (accessed 21 April 2010)