Project No. 249024

NETMAR

Open service network for marine environmental data

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

ICT - Information and Communication Technologies Theme

D7.8 Semantic Data Delivery and Processing Services Cookbook

Reference: D7.8_Semantic-Data-and-Processing-Service-Cookbook_r1_20120730

Due date of deliverable (as in Annex 1): M0 + 30
Actual submission date: 30 July 2012

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

Plymouth Marine Laboratory

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)	

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 Torget (oysteint@met.no)

Author(s)

- Peter Walker, PML, (petwa@pml.ac.uk)
- Michael Grant, PML (mggr@pml.ac.uk)
- Jorge de Jesus (jmdj@pml.ac.uk)

Document approval

- Document status: Revision 1
- WP leader approval: 2012-07-30
- Quality Manager approval: 2012-07-30
- Coordinator approval: 2012-07-30

Revision History

Issue	Date	Change records	Author(s)
Draft 1	2012-07-24	First draft of the report.	Peter Walker
Draft 2	2012-07-30	Incorporate review recommendations	Peter Walker
1	2012-07-30	Final release approved by coordinator.	Torill Hamre

Executive Summary

The NETMAR project will develop a pilot European Marine Information System (EUMIS) that will enable users to search, download and integrate satellite, in situ and model data from ocean and coastal areas. EUMIS will be a user-configurable system offering flexible service discovery, access and chaining facilities based on open and widely adopted web GIS standards.

To support smart search, EUMIS will use a semantic framework coupled with ontologies for identifying and accessing distributed data, such as near-real time, forecast and historical data, which are marked up using different, but semantically related, keywords. To support dynamic generation of new composite products and statistics suitable for decision-making, it will use web processing services that can be chained together to form workflows that perform a series of operations on input data chosen by the user.

As part of the project, NETMAR has developed a number of data and processing services to fulfil the needs of four pilots within different applications domains (sea ice monitoring and forecasting, oil spill drift forecasting and shoreline cleanup, marine ecosystem monitoring and forecasting, and development of Coastal Web Atlases). Within these pilots use cases have been identified that involve rapid development of custom workflows using existing services.

Semantically enabled discovery services can be used to find services that may be useful in a particular domain; once these services have been identified data and processing services must be combined in a compatible manner to produce the desired products. Work has been carried out within the project to extend services with semantic metadata (metadata specifying the meaning of a dataset and its units) allowing this chain building to be aided by an intelligent service chaining editor and for the processes themselves to verify their inputs.

The cookbook is primarily aimed at service providers/implementers who would also like to add semantic metadata to their services, and aims to provide guidance in the form of example XML and code snippets. It requires an understanding of the OGC WxS standards and XML. For those developing processing services familiarity with Python would also be useful.

This is not a proposal for a new standard. Instead it works within the existing OGC standards such as WMS, WFS and WCS, and services using this approach will remain compatible with clients that have no knowledge of semantics.

Contents

1. Introduction

1.1. What is metadata?

A good description of metadata and its uses is provided by the ICAN Semantic Interoperability Pilot Cookbooks[1] - Understanding Metadata, excerpts of which are reproduced below.

Geospatial metadata is “data about data”. It contains information that documents the basic characteristics of a geospatial data resource. It can also document basic characteristics of geospatial applications or services. Metadata falls into broad categories where it answers the “what, why, when, who, where and how” questions about the resource.

Discovery metadata helps a user to find or discover the data that they need and, thereafter, evaluate whether this resource satisfies the user’s requirements. Once a user has chosen the resource, usage metadata is then required to help fully understand and interpret the data.

Semantic metadata makes use of controlled vocabularies and thesauri to provide contextually relevant information allowing meaningful interpretation of the data. Strict control of the vocabularies makes it easier to construct automated methods of comparing and correlating diverse data. These thesauri are commonly represented using SKOS[2] (Simple Knowledge Organization System) and RDF[3] (Resource Description Framework).

1.2. Why do we need semantic metadata?

As processing systems become more complicated and datasets increasingly large, the need for mechanisms to automatically deal with this complexity becomes more and more urgent. We cannot rely on a highly trained user manually searching for useful datasets before downloading and processing them locally.

By adding semantic metadata to data and processing services it becomes possible to add a degree of “machine intelligence”, allowing the creation of tools that assist the user in finding data, creating bespoke workflows for their needs and protecting them against misusing data. The discovery aspect of metadata has been addressed elsewhere, this document will set out an approach to adding semantic metadata to OGC web services to enable the second two goals, simple workflow creation and semantically aware processing services (e.g. a temperature-oriented processing service that is intelligent enough to ensure it has relevant, compatible and correct input data representing temperatures).

A further use of semantic metadata is to represent uncertainty information related to a dataset. This differs from the earlier use as the uncertainty will normally be more complex and be linked to a specific data item rather than describing all data.

[1] D7.9.2 ICAN semantic interoperability pilot cookbooks, available from http://netmar.nersc.no/
[2] http://www.w3.org/2004/02/skos/
[3] http://www.w3.org/RDF/

1.3. OGC Standards

This cookbook deals with adding metadata to datasets published using OGC standards such as WMS, WFS and WCS. Other standards exist, such as OPeNDAP[4], which is widely used within the earth science community, and different approaches may be required in these cases. It is also important to point out that the suggestions made here are compatible with existing OGC standards, allowing interoperability with pre-existing services. An OGC discussion document, Semantic annotations in OGC standards[5], has been produced and the suggestions within this cookbook use this as a basis.

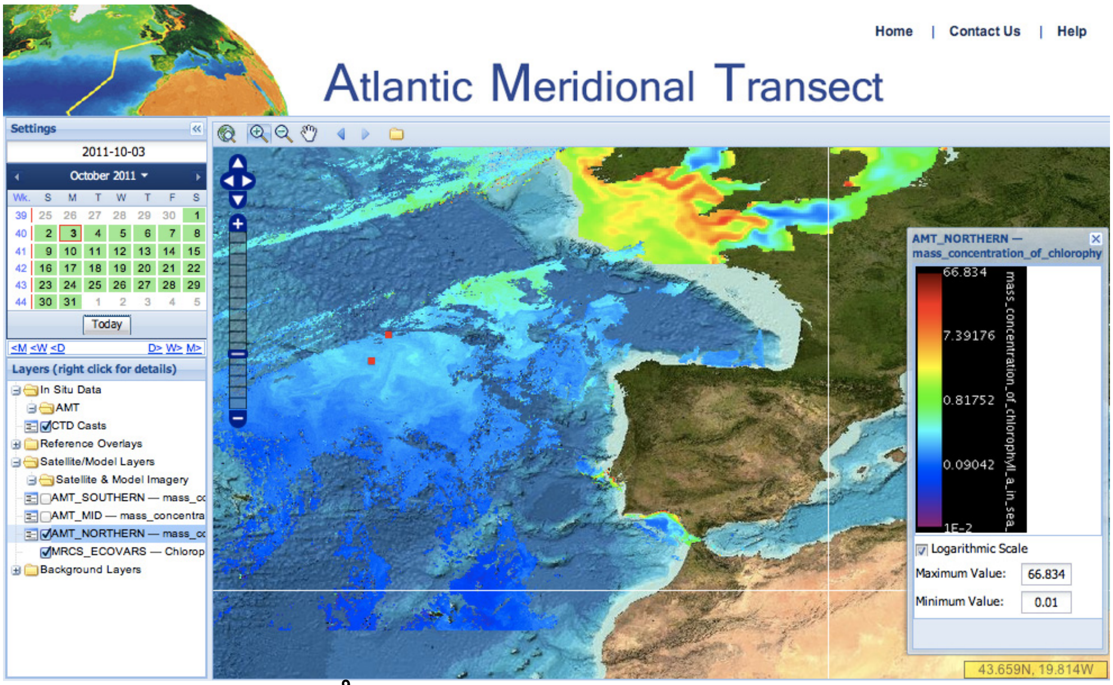
1.3.1. Web Data Delivery Services

The OGC has defined standards for a number of services delivering data over the web:

- WMS – The Web Map Service[6]
- WFS – The Web Feature Service[7]
- WCS – The Web Coverage Service[8]

1.3.1.1. Web Map Service

The OGC Web Map Service (WMS) standard supports the electronic retrieval of digital maps, which portray geographic data. A WMS server produces maps that are rendered in a pictorial format such as PNG, GIF or JPEG, or occasionally as vector-based graphical elements in Scalable Vector Graphics (SVG) or Web Computer Graphics Metafile (WebCGM) formats. WMS is important because it provides a means to transform raw spatial data into a portrayal that can be used within decision support tools. WMS makes it possible to display, navigate, zoom in, zoom out, pan, or overlay spatial datasets and to display legend information (Figure 1-1).


Home | Contact Us | Help
Atlantic Meridional Transect
Settings
Today
<M <W <D
D> W> M>
Layers (right click for details)
AMT_NORTHERN —
mass_concentration_of_chlorophy
66.834
7.39176
0.81752
0.09042
1E-2
mass_concentration_of_chlorophyll_a_in_sea_
43.659N, 19.814W

2011-10-03

October 2011

Wk.	S	M	T	W	T	F	S
39	25	26	27	28	29	30	1
40	2	3	4	5	6	7	8
41	9	10	11	12	13	14	15
42	16	17	18	19	20	21	22
43	23	24	25	26	27	28	29
44	30	31	1	2	3	4	5

- In Situ Data
  - AMT
    - ☑ CTD Casts
- Reference Overlays
- Satellite/Model Layers
  - Satellite & Model Imagery
    - □ AMT_SOUTHERN — mass_cc
    - □ AMT_MID — mass_concentra
    - ☑ AMT_NORTHERN — mass_cc
    - ☑ MRCS_ECOVARS — Chlorop
- Background Layers

☑ Logarithmic Scale

Maximum Value: 66.834

Minimum Value: 0.01

Figure 1-1 AMT Web portal[9] showing EO, Model and in situ data.

[4] http://opendap.org/
[5] http://portal.opengeospatial.org/files?artifact_id=34916
[6] http://portal.opengeospatial.org/standards/wms
[7] http://portal.opengeospatial.org/standards/wfs
[8] http://portal.opengeospatial.org/standards/wcs
[9] http://rsg.pml.ac.uk/gis/amt

ISO has approved WMS (version 1.3.0) as international standard ISO 19128:2005[10].

1.3.1.2. Web Feature Service

The OGC Web Feature Service (WFS) standard supports the electronic retrieval of geographic data as discrete “features”. Examples of features include a point feature (e.g. buoy), a line feature (e.g. a ship’s track), and a region feature (e.g. an oil slick). These features can have attributes, such as sea temperature along a ship’s track, which will be returned along with the geographic information. WFS clients may select features from a feature based on spatial constraints and any other criteria attributes. Unlike WMS which returns static maps in an image format, WFS returns meaningful data (including geographic information) in Geography Markup Language (GML). GML is an XML grammar defined by the OGC to express geographical features. WFS itself is a fairly mature standard supported by a number of tools. The current version is WFS 1.1.0 (also ISO 19142).

1.3.1.3. Web Coverage Service

The OGC Web Coverage Service (WCS) standard supports the electronic retrieval of geographic data as discrete "coverages". A common example of a coverage is a regular grid (e.g. a 2D raster dataset, a 4D gridded dataset, etc.). The WCS standard enables WCS clients to choose portions of a server's geographic data holdings based on spatial constraints and other criteria. Similar to WFS, WCS returns the original geographic data. The format of the geographic coverage data returned includes, among others GML, geoTIFF[11] and netCDF-CF[12]. Community schemas and conventions for retrieved data encodings are important to improve semantic data interoperability between WCS servers (e.g. application schemas for GML coverage, CF convention for netCDF format, etc). WCS 2.0 is currently under development but most current services support WCS 1.x.

1.3.2. Web Processing Services

In addition to the data delivery services the OGC has also defined a standard for delivering processing services of the web, WPS, the Web Processing Service[13].

The OGC Web Processing Service (WPS) standard provides a framework and rules for standardising inputs and outputs for geospatial processing services. A processing service may offer algorithms ranging from simple processes (e.g. calculations such as subtracting one set of spatially referenced numbers from another) to complicated processes (e.g. a global climate change model). The WPS standard also defines how a client can request the execution of a process, and how the output from the process is handled. It defines an interface that facilitates the publishing of geospatial processes and clients’ discovery of and binding to these processes. WPS 1.0.0 is the current version.

1.4. Target implementations

Many implementations of these standards exist, both open source and proprietary. In this cookbook we have used GeoServer[14] and MapServer[15] for WMS, WCS and WFS

[10] http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=32546
[11] http://www.remotesensing.org/geotiff/spec/geotiffhome.html
[12] http://cf-pcmdi.llnl.gov/documents/cf-conventions/1.6/cf-conventions.pdf
[13] http://www.opengeospatial.org/standards/wps
[14] http://geoserver.org

services although the metadata formats suggested conform to the OGC specifications and should be supported by all compliant implementations. Where this is not the case it will sometimes be possible to add an intermediate server which can provide this metadata whilst proxying data from the original server. For instance, GeoServer allows a remote WMS to be specified as a data source.

There are fewer WPS implementations available and, as it is a much less developed standard, more work may be required in order to implement the cookbook suggestions. We have chosen PyWPS for the examples in this cookbook. PyWPS[16] is a WPS implementation written in the Python language. The current stable version, 3.2.0, offers WPS 1.0.0 support. PyWPS provides a framework where programmers can deploy their geospatial algorithms and projects such as wps-grass-bridge[17] can be used to provide many processes “out of the box”. GRASS GIS (Geographic Resources Analysis Support System) is a free, open source geographical information system (GIS) capable of handling raster, topological vector, image processing, and graphic data[18], wps-grass-bridge makes these tools available through a web processing interface.

[15] http://mapserver.org

[16] http://pywps.wald.intevation.org

[17] http://code.google.com/p/wps-grass-bridge

[18] Neteler, M.; Mitasova, H. (2008). Open Source GIS : a GRASS GIS approach (3rd ed.). New York: Springer. ISBN 978-0-387-35767-6.

2. Requirements for semantically enabled services

A semantically enabled processing service will make use of semantic metadata provided for its data inputs and produce outputs that also contain semantic metadata. A number of requirements (in the form of metadata specifying the meaning of a dataset and its units) have been identified which must be fulfilled before semantic processing can take place. It should be emphasised that if these requirements are not fulfilled the processing service should fall back to a non-semantic processing approach, for instance not checking parameter types.

2.1. Data service metadata requirements

Data inputs to semantic processing services should provide semantic metadata such that the processing service can verify the type of data input parameters and their units. This is achieved by creating an RDF fragment containing this information (using standard vocabularies) and providing a link within the existing metadata to the fragment which could be stored on a separate server.

Methods for including this metadata, which remain compatible with the existing OGC WxS standards, are listed below. In all cases it should be noted that the metadata is optional and if it does not exist processing should fall thorough to the non-semantically aware version if possible.

2.1.1. Web Map Service (WMS)

In WMS the *GetCapabilities* operation return includes a *MetadataURL* element which can include a *OnlineResource* element pointing to the RDF fragment. In the simplest case all elements of this type would be checked to see if they linked to semantic metadata. Optionally the tag could be marked with the attribute ‘xlink:title="Semantic Metadata"’ and, if found, only elements with this attribute would be followed resulting in fewer network accesses.

```
<MetadataURL type="other">
  <Format>text/plain</Format>
  <OnlineResource xlink:type="simple" xlink:title="Semantic Metadata"
    xlink:href=" http://vocab.nerc.ac.uk/collection/P01/current/PSSTRT01"/>
</MetadataURL>
```

If it is not possible to use the *MetadataURL* element then the *Keyword* element may be used as a substitute. Use the URI as below.

```
<Keyword>
  http://vocab.nerc.ac.uk/collection/P01/current/PSSTRT01
</Keyword>
```

2.1.2. Web Coverage Service (WCS)

In WCS the *CoverageDescription* operation return includes a *metadataLink* element that can be used for this purpose. For instance metadata information added with the GeoServer web GUI interface would be shown as below:

```
<wcs:metadataLink metadataType="other">
  http://vocab.nerc.ac.uk/collection/P01/current/PSSTRT01
</wcs:metadataLink>
```

If it is not possible to use the *MetadataURL* element then the *Keyword* element may be used as a substitute. Use the URI as below.

```
  <Keyword>
   http://vocab.nerc.ac.uk/collection/P01/current/PSSTRT01
  </Keyword>
```

2.1.3. Web Feature Service (WFS)

In WFS the *DescribeFeatureType* operation returns metadata in the same way as for WMS, using *MetadataURL*.

```
 <wfs:MetadataURL type="other" format="text/plain" xlink:title="Semantic Metadata">
   http://vocab.nerc.ac.uk/collection/P01/current/PSSTRT01
 </wfs:MetadataURL>
```

If it is not possible to use the *MetadataURL* element then the *Keyword* element may be used as a substitute. Use the URI as below.

```
  <Keyword>
   http://vocab.nerc.ac.uk/collection/P01/current/PSSTRT01
  </Keyword>
```

2.2. Processing service metadata requirements

The processing service case (as implemented by an OGC WPS) is slightly more complex in that a WPS process can both provide outputs (similar to a data service) and also expect inputs. Luckily the *ProcessDescription* element allows *Metadata* elements within both *DataInputs* and *ProcessOutputs* elements which can be used for this. An example return to a DescribeProcess call is shown below.

```
<ProcessDescription wps:processVersion="0.1" storeSupported="true" statusSupported="true">
...
   <DataInputs>
      <Input minOccurs="1" maxOccurs="1">
...
        <ows:Metadata xlink:title="Degrees Celsius"
xlink:href="http://vocab.nerc.ac.uk/collection/P06/current/UPAA/"/>
...
      </Input>
   </DataInputs>
   <ProcessOutputs>
      <Output>
...
         <ows:Metadata xlink:title="Degrees Kelvin"
xlink:href="http://vocab.nerc.ac.uk/collection/P06/current/UPKA"/>
...
      </Output>
   </ProcessOutputs>
</ProcessDescription>
```

This shows that the process will accept its input in Celsius while its output will be in Kelvin. If the process expected a specific parameter type, for instance Sea Surface Temperature then this could also be specified here and invalid input (say Air Temperature) would be rejected even if they used the correct units (Celsius). We will elaborate on this usage later in the document.

3. Uncertainty metadata

3.1. *What is uncertainty?*

A brief description of uncertainty (extracted from the **NETMAR Semantic Framework Specification[19])** is included below.

In scientific data, error is commonly defined as the disagreement between a measurement and the real value. As error is never exactly known in practice, what scientists usually look for is **uncertainty**, which is defined as an interval of confidence around the measured value, within which the real (true) value is meant to lie. Obviously this is only an estimate of the error. It quantifies the expected accuracy, but it is not a guarantee of accuracy.

Uncertainty may be expressed as an absolute value (absolute uncertainty), for instance 100 ±5, or relative to the base measurement (relative uncertainty), for instance as a percentage, ±20%.

3.1.1. Global Uncertainty vs. Local Uncertainty

A geospatial dataset is a set of features (vector dataset) or pixels (raster dataset) described through a set of attributes (parameters). The uncertainty of the measurement of one of these attributes, a, for the dataset may be defined either globally as an aggregated value u(a) for the whole dataset, or locally as a value ux(a) for each feature or pixel x. The former is called global, or aggregated uncertainty, while the latter is called local or granular uncertainty. Figure 3-1 illustrates global and local uncertainties for a raster dataset.

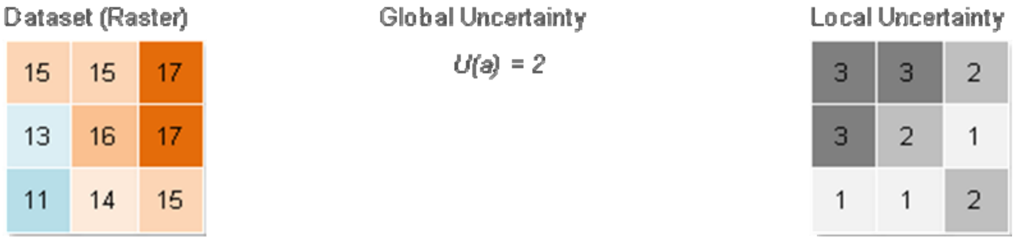


Figure 3-1 Global vs local uncertainty

Uncertainty metadata is handled using a similar approach to the other semantic metadata. However, whereas the semantic metadata for a layer in a WMS applies to all instances of that layer (for instance data for different days) the uncertainty information will vary depending on the actual data. To allow for this, the uncertainty metadata provides a rule by which the uncertainty information for a dataset may be retrieved rather than the uncertainty itself. The actual uncertainty information would be held elsewhere (probably with the data).

In general the metadata link will use the same format as described in Section 2.1 but the link could point to an XML snippet containing the uncertainty metadata rather than the uncertainty information itself.

For instance, a WCS entry might appear as below:

[19] NETMAR Semantic Framework Specification, Lassoued Y, et. Al. 2011.

```
<wcs:metadataLink metadataType="other" about="Uncertainty Metadata" xlink:title="Uncertainty Metadata">
    http://rsg.pml.ac.uk/projects/netmar/uncertainty/MODIS_SST_9km.xml
</wcs:metadataLink>
```

The XML snippet will contain the information necessary to retrieve the uncertainty information for a specific layer (uncertainty layer identifier and metadata).

In simple cases the uncertainty layer information may be encoded directly in the metadata link as is done for semantic metadata.

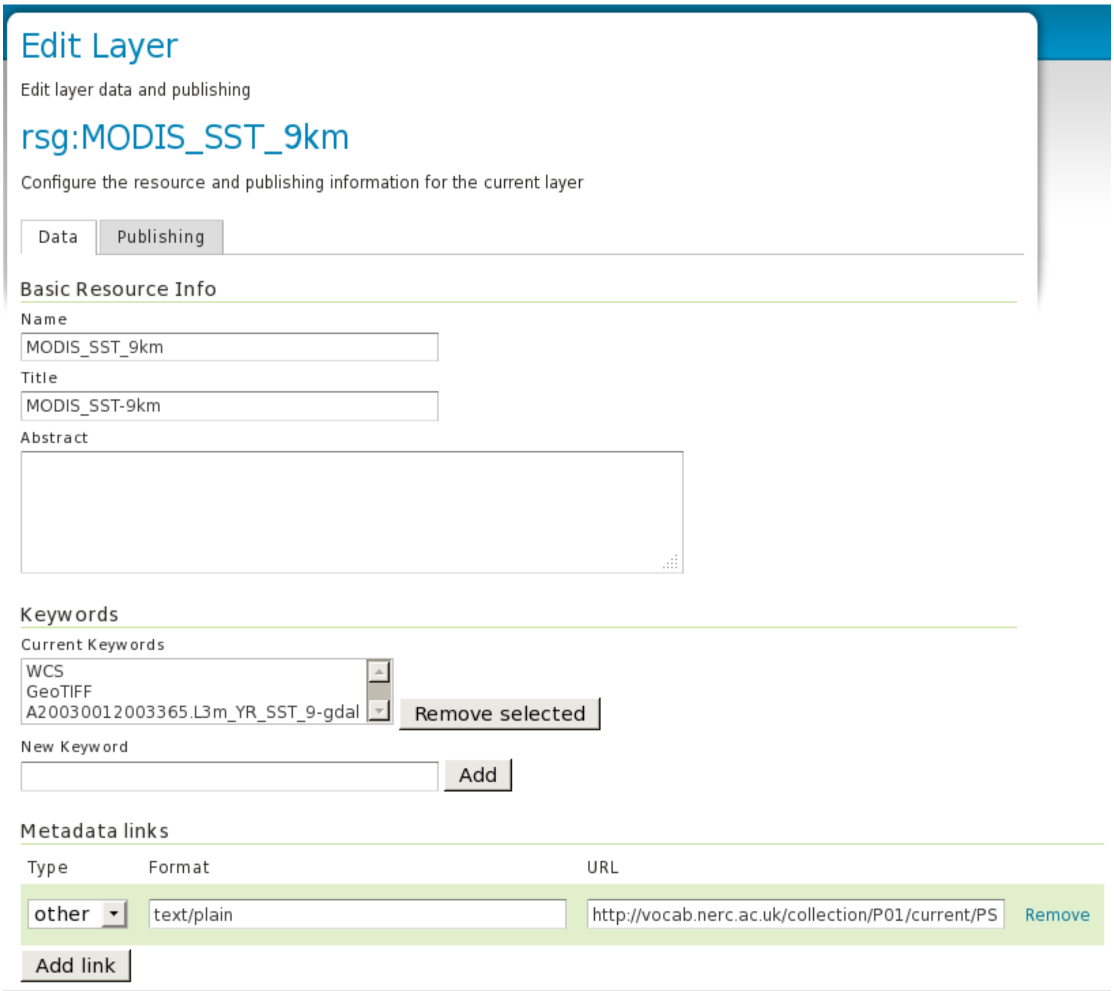
For instance, a WMS entry in this form would appear as below:

```
<wcs:metadataLink metadataType="other" about="Uncertainty Metadata" xlink:title="Uncertainty Metadata">
  http://rsg.pml.ac.uk/wfs?service=WMS&version=1.1.0&request=GetMap&layers=rsg:MODIS_SST_9km_variance
</wcs:metadataLink>
```

4. Implementation

4.1. Semantic metadata markup

Methods for marking up data services with semantic metadata will vary depending on the server implementation. For example, using GeoServer, the metadata for a WCS service may be entered using the web interface (Figure 4-1).



Edit Layer

Edit layer data and publishing

rsg:MODIS_SST_9km

Configure the resource and publishing information for the current layer

Basic Resource Info

Name: MODIS_SST_9km

Title: MODIS_SST-9km

Abstract:

Keywords

Current Keywords:
WCS
GeoTIFF
A20030012003365.L3m_YR_SST_9-gdal

Metadata links

Type	Format	URL	
other	text/plain	http://vocab.nerc.ac.uk/collection/P01/current/PS	Remove

Figure 4-1 Adding semantic metadata to a GeoServer layer.

The metadata created is shown below (WCS 1.0.0):

```
<wcs:CoverageDescription xsi:schemaLocation="http://www.opengis.net/wcs
http://rsg.pml.ac.uk:80/geoserver/schemas/wcs/1.0.0/describeCoverage.xsd" version="1.0.0">
  <wcs:CoverageOffering>
    <wcs:metadataLink metadataType="other">
      http://vocab.nerc.ac.uk/collection/P01/current/PSSTRT01
    </wcs::metadataLink>
    <wcs:description>
      Generated from GeoTIFF
    </wcs:description>
    <wcs:name>
      rsg:MODIS_SST_9km
    </wcs:name>
    <wcs:label>
      MODIS_SST_9km
    </wcs:label>
…
  </wcs:CoverageOffering>
</wcs:CoverageDescription>
```

To add semantic metadata to PyWPS processes is simply a matter of adding the attribute to each input and output definition within the initialisation of the process. For instance, to add semantic metadata to the inputs and outputs of the WPS example shown in Section 5 the following Python code (shown in bold) would be added.

```
self.inTemp = self.addLiteralInput("in", "Temperature input value", \
                        "Temperature input value that will be transformed from C into K",  \
                        type=type(0.0), \
                        metadata=[{'title': 'Semantic Metadata',  \
                                      'href': 'http://vocab.nerc.ac.uk/collection/P06/current/UPAA/'}], \
                        default=0.0)
self.outTemp = self.addLiteralOutput("out","Temperature output value", \
                         "Returned temperature in Kelvin", \
                         type=type(0.0), \
                         metadata=[{'title': 'Semantic Metadata', \
                                       'href': 'http://vocab.nerc.ac.uk/collection/P06/current/UPKA'}])
```

The addLiteralInput and addLiteralOutput functions are used in PyWPS to define the inputs and outputs of the process and values entered into the metadata parameter will be reflected in the process description as shown in section 5.2.

4.2. Unit and parameter compatibility verification

Once this semantic metadata has been defined for data and processing services how do we make use of it?

There are two suggested scenarios:

1. Add semantic checking to a service chaining editor which would be used to combine simple processes and data to produce more complex results. The editor would retrieve the semantic information and use it to ensure that inputs and outputs of the components of the service chain were compatible, perhaps suggesting transformation services to aid this.
2. Semantically aware processing services which verify the semantics of their inputs, either performing intelligent data transformations or raising exceptions if a mismatch is detected.

This checking requires that all services use a common vocabulary so that terms are used consistently. The NETMAR project has decided to use the NERC Vocabulary Server (NVS), maintained by BODC, to hold semantic metadata. The metadata is stored as RDF in a number of distinct vocabularies which may be accessed via HTTP GET. However, there is no requirement to use a specific vocabulary service, provided data and processing services that need to interoperate work from compatible vocabularies.

An example entry[20], which might be returned for an SST (Sea Surface Temperature) layer, is shown below:

```
<rdf:RDF>
  <skos:Concept rdf:about="http://vocab.nerc.ac.uk/collection/P01/current/PSSTRT01/">
    <skos:prefLabel xml:lang="en">Skin temperature of the water body by infra-red radiometer</skos:prefLabel>
    <skos:altLabel xml:lang="en">RadTmp</skos:altLabel>
    <skos:definition xml:lang="en">Unavailable</skos:definition>
    <dc:identifier>SDN:P01::PSSTRT01</dc:identifier>
    <skos:notation>SDN:P01::PSSTRT01</skos:notation>
    <owlxml:versionInfo>1</owlxml:versionInfo>
    <dc:date>2009-11-03 16:19:38.0</dc:date>
    <skos:note xml:lang="en">accepted</skos:note>
    <owlxml:deprecated>false</owlxml:deprecated>
```

[20] http://vocab.nerc.ac.uk/collection/P01/current/PSSTRT01

```
  <skos:broader rdf:resource="http://vocab.nerc.ac.uk/collection/P02/current/PSST"/>
  <skos:related rdf:resource="http://vocab.nerc.ac.uk/collection/P06/current/UPAA"/>
 </skos:Concept>
</rdf:RDF>
```

A processing service may check its inputs for compatibility by following these steps:

1) Retrieve the parameter RDF from the vocabulary server for each service.
2) Compare the "broader" links to the P02 collection ("parameter groupings" collection)
    a. if they are the same, the parameters are compatible, carry on to unit checking (step 3).
    b. if they are not the same, they are not compatible, return failure.
3) Compare the "related" links to the P06 collection (units)
    a. if they are the same, the units are directly compatible, return success.
    b. if they are not the same, return failure (or implement the unit compatibility checks in Section 4.3).

The check function has been implemented in a Python library[21], which allows the verification to be carried out by a single function call. This can be used directly within a PyWPS process or as a web service either via a REST interface or a WPS call (for incorporation in a workflow).

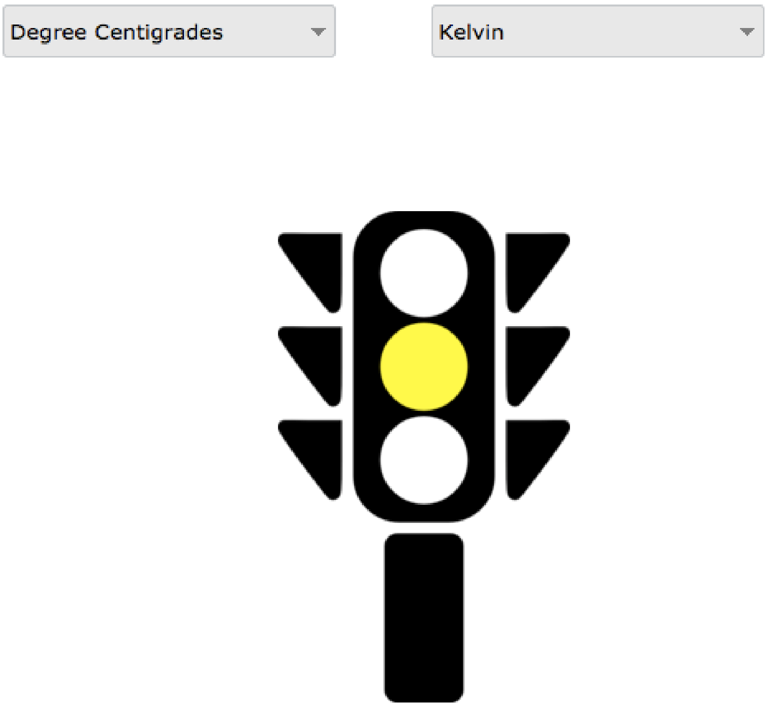
4.3. Unit compatibility and transformation

Some parameters may not be identical but it might be possible to map one to the other via a simple unit transformation. This could be carried out within a PyWPS process that was unit aware or off-loaded to a web service within a workflow.

The steps for carrying this out are listed below:

1) Retrieve the RDF for the two units from the vocabulary server.
2) Take the units text from each e.g. "kilometres per hour" and "miles per hour".
3) Canonicalise the units, transforming them to basic SI units, a scale factor and an offset (e.g. using the udunits2 program)
    a. "kilometres per hour" becomes "0.277777777777778 ms^{-1}".
    b. "miles per hour" becomes "0.44704 ms^{-1}".
4) if the SI units are not the same, the units are incompatible, return failure.
5) if the SI units are the same, the units are compatible.
6) check if the scale factors are the same and the offset zero; if so, they are directly compatible, return success.
7) if the scale factors and/or offsets are different, the units are compatible but need a transformation; return "yellow light/partial success" indicating this.
8) optionally, return transformation parameters.

[21] http://netmar.nersc.no



Degree Centigrades

Kelvin

Figure 4-2 Simple traffic light service comparing Celsius and Kelvin

5. Example services

5.1. Web Map Service (GeoServer)

An example WMS layer marked up according to these guidelines may be found at http://rsg.pml.ac.uk/wfs?service=WMS&version=1.1.0&request=GetMap&layers=rsg:MODIS_SST_9km&styles=&bbox=-180.0,-90.0,180.0,90.0&width=660&height=330&srs=EPSG:4326&format=image/png (Figure 5-1).

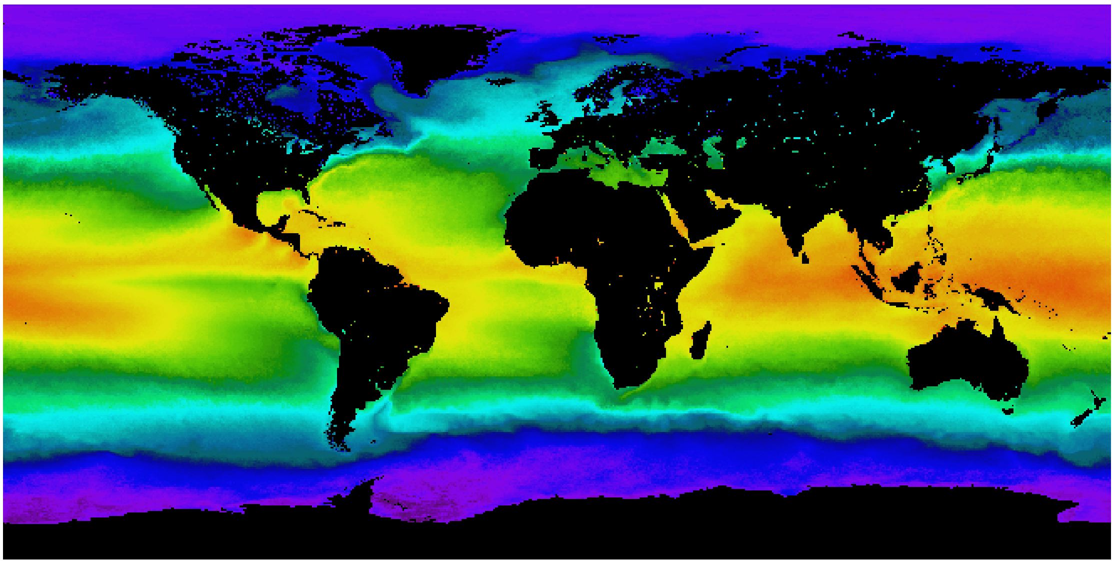


Figure 5-1 MODIS 9KM SST example layer.

A section of the GetCapabilities[22] response for this layer is shown below, with semantic metadata emphasised in bold:

```
   <Layer queryable="1">
     <Name>rsg:MODIS_SST_9km</Name>
     <Title>MODIS_SST-9km</Title>
     <Abstract/>
     <KeywordList>
       <Keyword>WCS</Keyword>
       <Keyword>GeoTIFF</Keyword>
       <Keyword>A20030012003365.L3m_YR_SST_9-gdal</Keyword>
     </KeywordList>
     <SRS>EPSG:4326</SRS>
     <LatLonBoundingBox minx="-180.0" miny="-90.0" maxx="180.0" maxy="90.0"/>
     <BoundingBox SRS="EPSG:4326" minx="-180.0" miny="-90.0" maxx="180.0" maxy="90.0"/>
     <MetadataURL type="other">
       <Format>text/plain</Format>
       <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink" xlink:type="simple"
xlink:href="http://vocab.nerc.ac.uk/collection/P01/current/PSSTRT01"/>
     </MetadataURL>
     <Style>
       <Name>raster</Name>
       <Title>Raster</Title>
       <Abstract>A sample style for rasters, good for displaying imagery</Abstract>
       <LegendURL width="20" height="20">
         <Format>image/png</Format>
         <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink" xlink:type="simple"
xlink:href="http://rsg.pml.ac.uk:80/geoserver/wms?request=GetLegendGraphic&amp;format=image%2Fpng&amp;wid
th=20&amp;height=20&amp;layer=MODIS_SST_9km"/>
       </LegendURL>
     </Style>
   </Layer>
```

[22] http://rsg.pml.ac.uk/wfs?service=WMS&version=1.1.0&request=GetCapabilities

5.2. Web Processing Service (PyWPS)

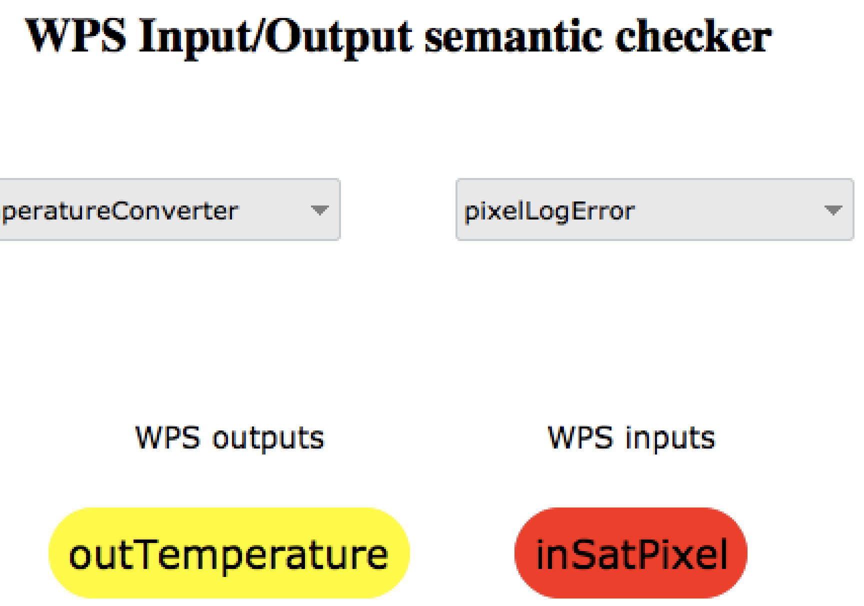
A simple WPS demonstrating how semantic metadata can be processed is available at http://rsg.pml.ac.uk/wps/generic.cgi?request=describeProcess&service=WPS&identifier=temperatureConverter&version=1.0.0

The results of the DescribeProcess are shown below:

```
<wps:ProcessDescriptions service="WPS" version="1.0.0" xml:lang="en-CA"
xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsDescribeProcess_response.xsd">
  <ProcessDescription statusSupported="true" storeSupported="true" wps:processVersion="0.1">
    <ows:Identifier>temperatureConverter</ows:Identifier>
    <ows:Title>Simple Temperature Converter, Centigrades to Kelvin</ows:Title>
    <ows:Abstract>Simple Temperature Converter, Centigrades to Kelvin</ows:Abstract>
    <ows:Metadata xlink:href="http://vocab.nerc.ac.uk/collection/P24/current/KELVIN" xlink:title="Temperature"/>
    <DataInputs>
      <Input maxOccurs="1" minOccurs="1">
        <ows:Identifier>in</ows:Identifier>
        <ows:Title>Temperature input value</ows:Title>
        <ows:Abstract>Temperature input value that will be transformed from C into K</ows:Abstract>
        <ows:Metadata xlink:href="http://vocab.nerc.ac.uk/collection/P06/current/UPAA/" xlink:title="Semantic
Metadata"/>
        <LiteralData>
          <ows:DataType ows:reference="http://www.w3.org/TR/xmlschema-2/#float">float</ows:DataType>
          <ows:AnyValue/>
          <DefaultValue>0.0</DefaultValue>
        </LiteralData>
      </Input>
    </DataInputs>
    <ProcessOutputs>
      <Output>
        <ows:Identifier>out</ows:Identifier>
        <ows:Title>Temperature output value</ows:Title>
        <ows:Abstract>Returned temperature in Kelvin</ows:Abstract>
        <ows:Metadata xlink:href="http://vocab.nerc.ac.uk/collection/P06/current/UPKA" xlink:title="Semantic
Metadata"/>
        <LiteralOutput>
          <ows:DataType ows:reference="http://www.w3.org/TR/xmlschema-2/#float">float</ows:DataType>
        </LiteralOutput>
      </Output>
    </ProcessOutputs>
  </ProcessDescription>
</wps:ProcessDescriptions>
```

This simple example process is defined to take an input temperature in Celsius (UPAA) and produce an output in Kelvin (UPKA). It can now check its input to ensure that it is flagged as using the correct units, for instance values from the data layer would be acceptable as when its semantic metadata is decoded it is found to use the same units. In addition a service chaining editor, such as that developed as part of the NETMAR project, can take advantage of this semantic metadata to ensure that data inputs to this process are flagged as using Kelvin and that any process using the outputs is expecting its input in Celsius.

A simple web application demonstrating the usage of this checking is available at http://earthserver.pml.ac.uk/wps/semantic/test2.html (Figure 5-2).



WPS Input/Output semantic checker

temperatureConverter ▾

pixelLogError ▾

WPS outputs

WPS inputs

outTemperature

inSatPixel

Figure 5-2 Demonstration semantic checking web application showing incompatible parameters.

6. Conclusions

This cookbook has shown that it is possible to add semantic metadata to data and processing services without breaking existing standards. Once services can provide and use metadata it becomes possible to build useful workflows which can check this metadata, both at creation and run time, and execute the processing accordingly. For instance a process which works upon temperature might detect, in real-time, if the input was supplied in units of Celsius, Kelvin or Fahrenheit and scale it accordingly. If it was provided with an incompatible input, such as height in metres, it would raise an exception rather than providing erroneous results. The ability to package uncertainty information with data raises a number of interesting possibilities such as passing the aggregate uncertainty down through a workflow or deliberately selecting data points with the lowest uncertainty.