European Environment Agency

Reportnet and metadata

A whitepaper

Introduction

This document describes how Reportnet uses metadata and how it associates it to a file or in more general terms, an object. Reportnet is object oriented, a file is only one kind of object. Other objects include obligations, countries, table definitions, deliveries, QA reports and events.

Reportnet deliveries

While Reportnet uses metadata in many areas, the core area is in the deliveries on CDR. A Reportnet delivery often consists of more than one file. We have therefore implemented a collection unit called a d*elivery envelope*. It holds the files, QA reports and the metadata describing what the delivery is for.

A delivery is a response to an obligation, and we have identified four properties that must be given to uniquely place a delivery in time and space.

1. Obligation identifier. It is used to link the delivery to an obligation. A delivery can have more than one obligation.
2. Spatial coverage. This is usually the country that reports, but can also be a region. Examples are Belgium, Gibraltar, The North Sea, the Pannonian bioregion.
3. Temporal coverage. Most obligations are periodic. The countries have to deliver e.g. yearly. It is therefore required to declare what timespan the delivery covers. It is given as a start date and an end date.
4. Release date. In case the provider redelivers a couple of months later, he'll have the same values for the above properties. The release date determines which delivery is the newest.

In addition to the four mandatory properties, there are a couple of optional ones. Title and description. They are not used by the system, but are for the convenience of the humans. Sweden uses the description property for its archival identifier.

Vocabularies

How does CDR know what obligation identifiers are possible? It calls ROD's webservice to get them. In other words, it uses ROD as a vocabulary. It gets not just titles, it uses obligation information to group the list on legislative instrument. The same happens with spatial coverage.

Files in the delivery

A file in a delivery has all of the mandatory envelope metadata plus some more:

- Content type expressed as an Internet media type. It is used to offer conversions to other formats.
- If the file is an XML file, then an XML schema identifier or DOCTYPE property is available. It is extracted from the file, but can also be set manually. It serves the same purpose as the content type, but for XML.
- Title
- Modification timestamp. It is used by the Reportnet system to remove QA reports that are obsolete with regards to the file the QA has run on.

QA reports

Reportnet has a system to do computerised quality assessments built into CDR. These are scripts specified by the client of a dataflow that tests whether the data respects certain requirements. The intention is to automatise some of the QA work and let the humans have time to do the complex QA.

When the automatic QA has been run on a file, the report is added to the file as metadata. There can also be manual QA reports. The only difference is that the information is added by a human operator.

QA reports sit somewhere between data and metadata. They are always associated with an envelope or a file, but they also have modification dates and authors. Reportnet treats them as data objects.

Handling metadata

CDR has an integrated metadata editor that serves its specific purpose and the metadata is stored in the CDR database as a record.

The user can zip an envelope, and thus can operate on the envelope as a single unit. If he does, a file called `metadata.txt` is *generated* containing the envelope's metadata. If there is going to be a SEIS specification for metadata files, it too could be generated.

Object discovery

The Reportnet Content Registry harvests CDR, ROD and several other repositories at regular intervals. These systems have arranged to have a *manifest file* located on a fixed location. The manifest describes what objects the site holds and the objects' metadata. Since Reportnet is object oriented it doesn't distinguish between files in CDR and obligations in ROD. They are all considered objects, but in ROD's case the manifest file contains most of the obligation data, whereas CDR only sends *conventional* metadata on the files. However, the format of the manifest is always the same. Reportnet uses the RDF specification.

Here is an example of *one* obligation in the ROD manifest file (shortened):

```
<rod:Obligation rdf:about="http://rod.eionet.eu.int/obligations/171">
  <dc:title>3 yearly report on quality of water for human consumption</dc:title>
  <dcterms:abstract>EC Water Reporting Obligation</dcterms:abstract>
  <dcterms:modified>2008-09-22</dcterms:modified>
  <dcterms:valid>2007-05-09</dcterms:valid>
  <rod:terminated>0</rod:terminated>
  <rod:eea_primary>0</rod:eea_primary>
  <rod:comment>Directive 98/83/EC has repealed Directive 80/778/EC with effect from five years after the entry into force (i.e. on 25.12.2003).</rod:guidelines>
  <rod:instrument rdf:resource="http://rod.eionet.eu.int/instruments/545"/>
  <rod:guidelines_url>http://ec.europa.eu/environment/water/water-drink/pdf/2007_05_09_guidance_doc_reporting.pdf</rod:guidelines_url>
  <rod:issue rdf:resource="http://rod.eionet.eu.int/issues/10"/>
  <rod:issue rdf:resource="http://rod.eionet.eu.int/issues/15"/>
</rod:Obligation>
```

Reportnet solves the issue of putting both database records and file metadata in the same Content Registry in an interesting way. It gives everything a globally unique URL. For a file it is the URL the file can be found at a website. For a ROD obligation, the system constructs a URL from the primary key and a prefix. In ROD's case it is http://rod.eionet.eu.int/obligations/171, which means obligation #171. ROD does the same with the issues. There is a small vocabulary of categories known as issues in ROD. Issue #10 is "Chemicals". The issues are also treated as (small) objects by Reportnet. ROD therefore constructs a URL from the primary key and a prefix. "Chemicals" becomes "http://rod.eionet.eu.int/issues/10".

Now we have an object in the Content Registry with the URL "http://rod.eionet.eu.int/issues/10". When getting the metadata for an obligation, relevant metadata includes what issues the obligation is linked to. Rather than writing "Chemicals" in plain text, ROD points to the URL of the object for the issue

“Chemicals”; http://rod.eionet.eu.int/issues/10. You can see the mechanism in the example above as the <rod:issue> elements.

Why is this useful? It makes it possible to attach additional data to the issue/category. You could add an explanation, links to narrower categories, synonyms and not least, other languages. A German could see “Chemikalien” and a Bulgarian “Химикал”. And finally, Reportnet knows about the possible values of the issues and can use it as a vocabulary.

If you would like to see the ROD example in the Content Registry, please go to:

http://cr.eionet.europa.eu/view_detail.jsp?description_id=c4f78cf5b54a3804114b2b9000a65062

As mentioned, CDR uses ROD as a vocabulary. The mechanism is exactly the same. Here is an example of a CDR delivery in the manifest file:

```
<rod:Delivery rdf:about="http://cdr.eionet.europa.eu/sk/eu/summerozone/envsqbjca">
  <dc:title>Summer ozone report 2008</dc:title>
  <dc:date>2008-10-23T11:47:32Z</dc:date>
  <dc:identifier
rdf:resource="http://cdr.eionet.europa.eu/sk/eu/summerozone/envsqbjca"/>
  <rod:obligation rdf:resource="http://rod.eionet.eu.int/obligations/386" />
  <rod:locality rdf:resource="http://rod.eionet.eu.int/spatial/33" />
  <dc:coverage>2008</dc:coverage>
</rod:Delivery>
```

Notice that the <rod:obligation> element points to http://rod.eionet.eu.int/obligations/386 and <rod:locality> points to http://rod.eionet.eu.int/spatial/33 respectively. These are objects for “Summer ozone exceedances “ and “Slovakia”.

Notifications

Reportnet has a unified notification system called UNS. It is based on subscriptions, in much the same way people subscribe to a mailing list. UNS is a central webservice, which receives notifications and then distributes them out to the subscriber via various mechanisms, of which email is only one.

The benefit is that the other systems don’t need to know who is subscribed to what. When there is something to inform about; a *single* notification is sent to UNS. The sender doesn’t need to know how it is disseminated.

Where does metadata enter the picture? The user can *filter* the notifications he wish to receive. If the user subscribes to notifications about upcoming deadlines, he can specify he only wants information relevant to Germany. When ROD generates such a notification, it creates a data object called an “Approaching deadline” event. There are typically multiple countries that have to report on the same date, so the list of countries to report on the obligation is added to the deadline event as metadata and sent to UNS. A person interested in German deadlines will have a filter in UNS saying country=”Germany”.

The mechanism is used everywhere. A delivery to CDR generates a notification that sends the envelope metadata as an “Envelope release” event. Dataflow clients filter on one or more obligations, countries on their own country name.

Here is an example of the metadata attached to an “Envelope release” notification:

Title	Envelope BW Report 2008 - Finland (http://cdr.eionet.europa.eu/fi/eu/bathing/envst93ua) released to public
Date	2008-Dec-10 09:18:22
Identifier	http://cdr.eionet.europa.eu/fi/eu/bathing/envst93ua
Obligation	Bathing Water Directive 76/160/EEC Report
Locality	Finland
Event_type	Envelope release
Actor	zacheout

As for manifests, the metadata sent to UNS also uses the RDF file format.

Introduction of SEIS into Reportnet

The way Reportnet intends to implement SEIS is to do more of the same. The main difference is that the participating nodes are not vetted, but can be any organisation. The second difference is that we are not just talking deliveries any longer, but data in a wider context.

Reportnet uses the obligation identifier, temporal and spatial coverage to categorise a dataset. If the data isn't related to an obligation any longer, then what do we do? In the absence of a SEIS standard on metadata, Reportnet takes an promiscuous approach. It accepts all metadata. The organisation creating the manifest file can put any metadata property in it. RDF allows that kind of thing.

But there are certain types of metadata that the provider can't add. Remember the QA reports? These are created by the *user* of the data. They can be stored on CDR, because CDR is the data user's repository and controlled by the *user.* When data is stored at the data provider's location, how do we store the QA reports? How do we arrange authorisation for these users to add QA reports, if it is "use by many" and we don't know who'll use the data?

In a wider context, you can easily think of more user-provided metadata. It is often the most useful. Look at Amazon. A major factor in Amazon's success is the user-supplied metadata on the books. Reviews, Categorisation, Users of this dataset also used X, Product Y uses this dataset, an so on.

Rather than adding requirements on SEIS nodes to accommodate reviewers, Reportnet will introduce a new site called QAW to store user-supplied metadata. Users can find a dataset using Reportnet's search engine or Google. The dataset will have a URL, which will be used as the key to associate additional metadata to. Consider an organisation called NGO.org having made a dataset available at the URL http://www.ngo.org/migrationpatterns.xml. Being a colaborating SEIS node, they provide Dublin Core metadata for the file in a manifest file where the record looks like this:

```
<rdf:Description rdf:about="http://www.ngo.org/migrationpatterns.xml">
  <dc:title>Migration patterns of populations in Europe 1990-2008</dc:title>
  <dc:date>2008-03-01</dc:date>
</rdf:Description>
```

The dataset is reviewed and a QA report is stored on QAW with a link to the URL. Below is shown what the manifest file for QAW could contain about the URL.

```
<rdf:Description rdf:about="http://www.ngo.org/migrationpatterns.xml">
  <qaw:review>This dataset is useful for a view on European level, but is too
granular to allow country-level analysis, such as rural to urban migration.
Provider organisation is against open borders, and data can be skewed.</qaw:review>
</rdf:Description>
```

It is not shown here who wrote the review. This is metadata on the metadata, and can be modelled also.

When Content Registry has harvested NGO.org, QAW and potentially other QAW-like sites, it will bring together all the metadata bits, allow search and show them together on a factsheet for the URL.

Mapping properties

A potentially significant issue in SEIS is how to handle different coding standards for metadata. One organisation codes a description as <description>, another as <abstract>. Or <title> is used for two different concepts; a document title and a person title.

Taking the last issue first, RDF (and hence Reportnet) forces you to use globally unique tags for your metadata via namespaces. Certain tags have been standardised. For instance Dublin Core has standardised about 15 tags for documents. You should therefore always use <dc:title> for a document and never for a person. If you can't find a predefined tag for your metadata item, you must invent your own globally unique tag. In fact, the same way as Reportnet uses URLs to represent objects, the metadata tags are also URLs. The Dublin Core <dc:title> tag is really fully written as `http://purl.org/dc/elements/1.1/title` and you can *treat* it as an object. You can add business logic to it.

Example: Someone has tagged his dataset's title as

```
<rdf:RDF xmlns:my="http://ngo.org/namespaces/">
<my:datasetname>Production of tyres</my:datasetname>
</rdf:RDF>
```

The way you declare the mapping between `http://purl.org/dc/elements/1.1/title` and `http://ngo.org/namespaces/datasetname` is as business logic is to describe with the RDF class mechanism that one is the subproperty of the other.

```
<rdf:Description rdf:about="http://www.ngo.org/namespaces/datasetname">
<rdfs:subPropertyOf rdf:resource="http://purl.org/dc/elements/1.1/title"/>
</rdf:Desription>
```

It doesn't have to be done by an administrator. You can do it yourself on QAW, and the effect will be system-wide. When you then search in Content Registry for *title* contains "*tyres*", the NGO.org dataset will be found. Others who disagree with your mapping can turn it of for themselves. The mechanism is available but the exact user-interaction is yet to be determined.

Diggining into files

As shown above, Reportnet's Content Registry is object oriented. It deals with both files and database records, and when it works with records it can follow references from one object to another. In the example of ROD, it is used a vocabulary linking deliveries to obligations.

What we will do with the SEIS datasets is that if the dataset can be understood, then it will be treated as more than just a file. The records in the dataset will be imported into the system as objects. We intend to do it first with XML files. As part of the agreement on the file format, a stylesheet translation script will be developed to facilitate the import. By doing it we're expanding the ROD mechanism to dataset with the same benefits; the ability to treat stations, NUTS codes, sites etc. as vocabularies other objects can link to.

But the biggest benefit is outside the scope of this document on metadata. It is that if you import the same type of data from several locations. e.g. countries, and convert them into objects, you will in practice have *aggregated* the data, which until now has been a very manual task. Hopefully not much longer.