Ocean Sci. Discuss., 12, 2655–2682, 2015
www.ocean-sci-discuss.net/12/2655/2015/
doi:10.5194/osd-12-2655-2015
© Author(s) 2015. CC Attribution 3.0 License.

Ocean Science Discussions
Open Access
EGU

This discussion paper is/has been under review for the journal Ocean Science (OS).
Please refer to the corresponding final paper in OS if available.

Technical note: Harmonizing met-ocean model data via standard web services within small research groups

R. P. Signell[1] **and E. Camossi**[2]

[1]USGS Woods Hole Coastal and Marine Science Center, Woods Hole, MA, USA
[2]NATO STO Centre for Maritime Research and Experimentation, La Spezia, SP, Italy

Received: 17 July 2015 – Accepted: 20 July 2015 – Published: 5 November 2015

Correspondence to: R. P. Signell (rsignell@usgs.gov)

Published by Copernicus Publications on behalf of the European Geosciences Union.

Abstract

Work over the last decade has resulted in standardized web-services and tools that can significantly improve the efficiency and effectiveness of working with meteorological and ocean model data. While many operational modelling centres have enabled query and access to data via common web services, most small research groups have not. The penetration of this approach into the research community, where IT resources are limited, can be dramatically improved by: (1) making it simple for providers to enable web service access to existing output files; (2) using technology that is free, and that is easy to deploy and configure; and (3) providing tools to communicate with web services that work in existing research environments. We present a simple, local brokering approach that lets modelers continue producing custom data, but virtually aggregates and standardizes the data using NetCDF Markup Language. The THREDDS Data Server is used for data delivery, pycsw for data search, NCTOOLBOX (Matlab®[1]) and Iris (Python) for data access, and Ocean Geospatial Consortium Web Map Service for data preview. We illustrate the effectiveness of this approach with two use cases involving small research modelling groups at NATO and USGS.

1 Introduction

Efficient and effective access to meteorological and ocean model data is essential for a wide range of applications, from the global climate projects to regional oceanographic issues. While systems such as Earth System Grid (Bernholdt et al., 2005) are in place for global climate projects, there are often no existing systems to search for and work with model data produced by smaller research groups. While these research groups may not produce the 10s of Petabytes of data produced by the climate community, they still often have significant holdings in the 10–100 TB range, too much to be delivered

[1]Mention of trade names or commercial products does not constitute endorsement or recommendation for use by the US Government.

OSD
12, 2655–2682, 2015

and handled by regional and national data centres. As they already store the data for local use, it makes sense to also serve these data to the public using standardized web services. Even for data that are not meant for public consumption, use of a private service-based approach can make working with data within a research group more effective. These data typically come from models using different formats and conventions, and a harmonization effort allows the development of common tools that can access data from any model, removing the need for model specific code. This makes it possible for researchers to spend less time on routine data tasks, and more time doing science.

The capabilities of distributed, federated data systems for both model and observational data have been continuously improving over the last decade. The Global Earth Observation System of Systems (GEOSS) portal[2] enables access to Earth observations collected worldwide and services for environmental research focusing on societal benefit areas such as agriculture, biodiversity, climate, disasters, ecosystems, energy, health, and water. Copernicus[3], the European contribution to GEOSS, is an Earth observation system for high quality harmonized data and models for land, marine, atmosphere, climate change, emergency management, and security issues. The US Integrated Ocean Observing System (IOOS) is another such system, directly contributing to the Global Ocean Observing Sytem (GOOS), the marine component of GEOSS.

These systems have been developed to provide harmonized access at regional, national and global levels to data and models for understanding the physical environment and to monitoring its evolution. The data discovery and access services they provide facilitate the distribution of data among different communities. A brokering approach to data harmonization leaves resource providers free to use the model and data format they prefer, yet allows their data to be accessible via standard community web services.

Indeed, this approach, where data are harmonized downstream of the service provider, has been historically more successful than top-down approaches imposing

[2]http://www.geoportal.org/.

[3]http://www.copernicus.eu/.

OSD
12, 2655–2682, 2015

the adoption of particular data formats or conventions. The adoption of open and standardized services facilitates model data access and reuse. This is the strategy for model interoperability adopted by IOOS (Signell, 2009; Signell and Snowden, 2014), which standardizes data models and services for ocean models across national and regional modelling centres by allowing a brokering approach to be implemented at the provider institution. This type of system, if made easy to install and learn, can be used effectively by small research groups.

Here we present two use cases where the IOOS system is applied to modelling data which are primarily intended for in house use, and only certain datasets are meant to be accessible to the outside world. Even within the group, the system enables search capability across multiple investigators and collaborators datasets and uniformity of access, which empowers and simplifies scientific analysis workflows. It also allows a standard way to make data accessible (with metadata) to the world, which can meet data publishing requirements required by government agencies as well as plugging into an international network of data providers.

We first describe the model data strategy used by IOOS, then describe some techniques and tools we have developed to make this strategy easier and more effective for smaller research groups. After presenting the use cases we discuss lessons learned and the need for future work.

2 The IOOS model data interoperability system

The IOOS model data infrastructure is built around community and international standards for data models and web services (de La Beaujardière et al., 2009; Howlett et al., 2014). The infrastructure requires that gridded data be delivered via the OPeNDAP service with Climate and Forecast (CF) Conventions, and for images of data, via Open Geospatial Consortium (OGC) Web Map Service (WMS). OPeNDAP is used because OGC services are not currently capable of delivering CF compliant model data in a standardized way.

OSD
12, 2655–2682, 2015

Although a variety of tools can be used to deliver OPeNDAP and WMS services, one convenient method is to utilize the THREDDS Data Server (TDS) from Unidata[4] as a local broker, turning collections of non-standard data files from modelers into aggregated, standardized datasets delivered by web services (Signell, 2010; Signell and Snowden, 2014). The transformation happens virtually, using NetCDF Markup Language (NcML), with XML templates created by specialists with knowledge of the standards, and then simply modified by modellers to suit their output. The data are represented internally in the TDS with a common data model aligned with the CF Conventions which delivers the data through a variety of web services, including OPeNDAP, WMS and NetCDF Subset Service. It also includes a ncISO service that generates an ISO 19115 compliant metadata record for each dataset based on the attributes in the input files and the NcML.

In addition to tools to provide standardized data distribution and access, it is also useful to have tools that allow investigators to search for datasets based on geospatial extent, time range, keywords and other descriptors (e.g. variables of interest). There are many tools that enable searching across metadata that describe distributed geospatial datasets: GeoNetwork opensource, Geoportal Server, GI-cat, CKAN, DKAN and pycsw. In particular, Python-based CKAN[5] and its Drupal-based competitor DKAN[6] are widely used to implement governmental open data portals (see for instance data.gov, data.gov.au, data.gov.uk). These tools provide off-the-shelf ready-to-use solutions for efficient data discovery. The CKAN search interface includes Google-like fuzzy search over keywords, faceted filtering for dataset browsing, and, if the spatial extension is enabled, geographic filtering and preview of datasets. It includes an OGC Catalogue Service for the Web (CSW) harvester and spatial datasets can be queried through the same standard, for example using pycsw, a Python implementation of the OGC

[4]THREDDS Data Server, available at http://www.unidata.ucar.edu/software/thredds/current/tds/.

[5]CKAN, The open source data portal software, available at http://ckan.org/.

[6]DKAN, http://nucivic.com/dkan/.

OSD
12, 2655–2682, 2015

CSW. The interface and the underlying catalog engine can be customized to achieve more powerful data discovery solutions. However, its customization through extensions development may be too technically challenging for smaller research groups.

3 A procedure for small research groups

The following is an approach for enabling model data interoperability for small research groups, utilizing procedures and a collection of technologies developed over a period of several years within the US IOOS program. We describe the components in the following order: data *delivery*, *access*, *search* and *preview*. We then present two test cases where this approach has been used, at the NATO Science and Technology Organization, Centre for Maritime Research and Experimentation (STO-CMRE) and the USGS CMG Sediment Transport Modeling Group.

3.1 Data delivery

Delivery of data is accomplished with the THREDDS Data Server (TDS). The first step is to install the TDS, a Java Servlet application which can be installed in a few hours following the TDS administration tutorial[7]. The installation is a cookbook procedure not involving special knowledge or skills on the part of the system administrator. Once installed, the TDS is configured to dynamically scan targeted directories for data of specified types. For example, the directory "/data/shared" could be scanned for files ending with ".nc, .cdf, .grib, .grb and .ncml". This means that any file of these types that are placed in this directory, or in a subdirectory below this directory, are immediately accessible via the TDS.

With the TDS configured with a *datasetScan*, modelers can create a subdirectory for their simulation, deposit the collection of NetCDF (or GRIB) files that make up the simulation, and then add a single NcML file. The NcML is an XML file that contains

[7] http://www.unidata.ucar.edu/software/thredds/current/tds/tutorial/index.html.

OSD
12, 2655–2682, 2015

the information the TDS uses to virtually aggregate the data and add or modify metadata to meet the CF Conventions. When this NcML file is accessed through the TDS, the aggregated standardized dataset can then be accessed through all the TDS web services, including OPeNDAP, NetCDF Subset Service, WMS and the ncISO metadata service. This NcML file approach is convenient for model data providers as no reload or restart of the TDS is necessary. This is clearly an advantage, as there is often a large collection of hindcasts, simulations testing the sensitivity to changes in parameterizations, data assimilation techniques, boundary conditions, etc.

For forecast models, a different approach is needed, and the NcML must be added to static catalogs. While new forecast data can be added to forecast aggregation without reloading the server, any modification to the static catalogs requires a restart of the TDS. This is usually not a major issue as groups typically only have one or two forecast models running.

3.2 Data access

Typically, users access the data directly from their developer environment, e.g. Matlab® or Python, using the OPeNDAP service. If if the extracted data are to be used repeatedly, however, a local copy of the extracted data can be saved as a NetCDF file using the TDS NetCDF Subset Service. Fortunately, both the Matlab® and Python tools can open a remote OPeNDAP dataset or a local NetCDF file using exactly the same syntax, so users need only to learn one syntax for extracting data.

For Matlab®, NCTOOLBOX[8] provides support for the CF Conventions by leveraging the Unidata NetCDF-Java library behind a simple Matlab® interface. For Python, the

[8]NCTOOLBOX, A Matlab® toolbox for working with common data model datasets, available at http://nctoolbox.github.io/nctoolbox/.

OSD
12, 2655–2682, 2015

Iris package[9] from the British Met Office provides support for the CF Conventions by leveraging the Unidata NetCDF C library behind a Python interface.

3.3 Data search

We use pycsw[10], which can be installed in minutes using the defaults settings, for cataloging the datasets and providing a standardized search capability. It ingests ISO 19115 compliant metadata, and can be controlled via simple command line arguments, providing easy scripting capability.

To control which datasets are harvested, we run a Python script that crawls specific local or remote THREDDS catalogs, and enables filtering on a dataset by dataset level (see the code in Fig. 1). For example, we can specify for a specific THREDDS catalog that metadata from .ncml files should be harvested, but data from .nc files should be ignored. For hindcast datasets configured with the *datasetScan* approach described previously, this allows only the aggregated datasets to be harvested, and the underlining finer datasets to be ignored. For forecast datasets that update periodically, we can run the Python script regularly using a scheduler (e.g. cron) so the dataset metadata are updated accordingly.

3.4 Data preview

The Open Geospatial Consortium Web Map Service (WMS) included in the THREDDS Data Server is provided by the ncWMS package (Blower et al., 2013). The TDS also includes the Godiva2 WMS client which allows for simple preview of individual datasets (Blower et al., 2009).

[9]IRIS, A Python library for Meteorology and Climatology, available at http://scitools.org.uk/iris/

[10]Pycsw, http://pycsw.org/.

OSD
12, 2655–2682, 2015

4 Use cases

We now describe two specific instances where the approach was implemented, discussing the issues and lessons learned, in order to improve the approach.

4.1 Centre for Maritime Research and Experimentation (CMRE)

The Centre for Maritime Research and Experimentation (CMRE) of NATO Science and Technology Organization is a research institution active in ocean science, underwater robotics, optics and acoustics. Most researchers are on temporary assignments (typically 3–7 years) from the 28 NATO countries. CMRE regularly conducts focused field trials in various regions of the ocean, involving multiple Nations and research institutions, running experiments integrating data collected from heterogeneous ocean observing networks that include several different observational assets, from in situ sensors to remote sensing, as well as numerical models.

During these field trials, real-time data are ingested in the CMRE scientific data management infrastructure, which has been developed to (1) facilitate data dissemination and sharing among CMRE scientists and partners, and to (2) achieve data interoperability relying on open services (Open Geospatial Consortium – OGC) and open software by enforcing compliance to NATO (e.g. Allied Geographical Publications, STANAGS) and international standards (ISO , OGC). A continuous deployment service integrating Python scripts, a web-based scheduler (Jenkins) and versioning software is used to automatically extract metadata from datasets, and to populate a data server (GeoServer instance) and the CMRE data catalog, which combines GeoNetwork and CKAN to realize an efficient and user friendly data discovery service. Newly acquired observations and data products become available to CMRE researchers in near real-time through the CKAN interface (see Fig. 5).

During and after the field trial, models are compared with each other and with observational data to assess and to improve model predictive skill. Simulations typically include a collection of many NetCDF files. In the past, the habitual research workflow

OSD
12, 2655–2682, 2015

involved the storage of various model runs on local hard drives, and the access to model outputs with model-specific routines, mainly written in Matlab®, for visualization and analysis. Exchange of data with other scientists was mainly done via FTP or via exchange of physical media.

To advance beyond this situation and to demonstrate the benefits of using the IOOS approach to data interoperability, we developed a prototype implementation and tested it against data from a specific field trial, REP14-MED, which was conducted in 2014 in the Sardinian Sea. We installed a THREDDS data server, and setup NcML for each of the collection of files that represented a REP14-MED model simulation. Although a different NcML file is required for each collection of files, NcML files for a particular type of model (e.g. ROMS), differ only in the specifics of the simulation (e.g. title, abstract, principal investigator, etc), making it possible to use an example NcML file as a template for other simulations. In this way an expert in standard conventions can create the initial NcML file, but then hand it off to modelers for them to copy, modify and use on their own. NcML transformations were used also to enforce the CF Conventions compliance of glider data. The transformation was applied on the fly when accessing the data through OPeNDAP. No modification of the original files was necessary, and the original data generation workflow was not affected in any way. Likewise, legacy models and software using the data can continue to access them using the original formats. This is a key advantage of this software, in particular when, as in our case, such data are already used as an input source of information by several users and services.

We installed the pycsw server using the default settings, which deploys a local SQLite database. We then deployed a custom python script (see Fig. 1) to crawl specified THREDDS catalogs and generating ISO 19115 compliant metadata that can be harvested with pycsw.

With the datasets standardized via NcML, accessible via OPeNDAP, and ISO 19115 compliant metadata that can be published via pycsw, the last task needed was to develop a simple CSW query capability for Matlab® users. Python users had the powerful

OSD
12, 2655–2682, 2015

OWSLib[11] library, but nothing existed for Matlab®. To meet this need, we developed a simple CSW query function for Matlab® that takes a bounding box, temporal extent, and a keyword on input, and retrieves all OPeNDAP data URL that match the criteria.

To test the approach, we first extracted sea surface salinity from all REP14-MED models in the CMRE data catalog. We accomplished this by formulating a geospatial and temporal CSW search for the REP14-MED region and time period in Matlab®, including a search for the text string "sea_water_salinity", the CF Standard Name for salinity. This search returned metadata records from four datasets that matched the criteria, and then obtained the OPeNDAP endpoints from the metadata records. We then opened each endpoint in Matlab® using NCTOOLBOX and extracted the sea surface salinity data for a specific time during the REP14-MED experiment (the results are shown in Fig. 3). Finally, we extracted virtual glider data from the four models along a glider track, using the NC_GENSLICE routine from NCTOOLBOX (the results are shown in Fig. 4), demonstrating that NCTOOLBOX was able to automatically determine the horizontal and vertical coordinates without model specific code.

Although Matlab® was the environment used by most modelers at CMRE, an IPython Notebook Server was also installed[12], so that CMRE researchers could explore standards-based access to the same datasets via Python. The IPython notebook was particularly convenient because it uses a client/server model, with a standard web browser as the client. This allowed researchers to test out Python browsing, access and visualization using only a browser, with no installation of software necessary on the local machine (cf. Figs. 5 and 6). The IPython notebook is also convenient as an instructional tool, with rich text, inputs and outputs, all existing on the same page that may be first inspected, and then run interactively.

The installation and configuration of the system took place while the first author was visiting CMRE for one month under the CMRE Visiting Researcher Programme. It was

[11] https://geopython.github.io/OWSLib/.

[12] IPython Notebook http://ipython.org/notebook.html was also installed.

OSD
12, 2655–2682, 2015

deemed successful enough that its deployment in the production environment is an official part of the CMRE work plan for 2015.

4.2 USGS Coastal and Marine Geology (CMG) Program Sediment Transport Group

USGS CMG scientists at the Coastal and Marine Science Center in Woods Hole, Massachusetts, USA conduct many ocean model simulations as part of their research to understand the behaviour and impact of suspended sediment in estuaries, coastal areas and regional seas. Many of these studies utilize the Coupled Ocean Atmosphere Wave and Sediment Transport (COAWST) model, and often many simulations are performed as part of a sensitivity study, varying different forcing and configuration parameters to determine their importance. Each simulation typically produces a collection of NetCDF files that are not completely CF Convention compliant.

As in the CMRE case, we used the THREDDS server configured with *datasetScan*, so that researchers could simply add NcML files that allowed the data to be accessed as standardized, aggregated datasets. Although NcML files can be edited with a text editor starting from supplied templates, the cut-and-paste manual editing of NcML files is prone to error, such as copying NcML from an existing simulation and not removing the attributes for a variable that was no longer being output. To solve this problem, we simplified the editing task by having modelers just to modify a simple YAML[13] input file which contained only simulation specific information. Then we provided researchers with a python script that reads the YAML input file and produces a valid NcML file. The python script was developed for the COAWST/ROMS models, but could easily be expanded to cover other model types.

Because modelers wanted selected model runs to be displayed in the official USGS CMG Portal, we provided a simple mechanism where they specify *CMG_Portal* as a project attribute in the metadata. This information is automatically included in the

[13] http://www.yaml.org/about.html.

OSD
12, 2655–2682, 2015

ISO metadata, and becomes queryable via the CSW service. An example of the YAML input file and the resulting output NcML file are shown in Figs. 7 and 8. Thus the portal developer (a 3rd party external to the USGS) need only to query the CSW service for records where the *project* attribute matches *CMG_Portal* and then utilize the WMS viewer to enable preview of the data in the browser (Fig. 9).

5 Discussion

The two use cases presented in the previous section proved valuable in several ways. By testing the approach with specific workflows and datasets, we identified gaps and weaknesses in the tested technologies, but also in the documentation, training and support material. All these factors are important for the successfull implementation of these approaches by the scientific community.

In large operational centres, mandates are possible, with all models driven by common workflows, producing files with common formats and conventions. The installation and configuration of the systems can be complex, supported by substantial IT infrastructure.

At smaller research institutions, researchers use individual workflows when producing model data and the models they use differ according to their research needs and experiences, resulting in variegate data formats and conventions. In addition, IT infrastructure supporting research is often overextended. It is not surprising that for small research groups to adopt a new approach, the approach must be easy to install, configure and maintain. It is also useful if the approach integrates into the workflows they already use: distributing data using files they already create, and searching for and accessing data in analysis systems they already use (e.g. Matlab®, Python). It was clear from both use cases, however, that documentation needs to be improved. Users both at CMRE and USGS requested more online documentation and instructional videos.

Both use cases demonstrate that there are many collateral benefits to the standardized approach, beyond improving the local delivery, access, search and preview. For

OSD
12, 2655–2682, 2015

smaller research groups, the automatic generation of ISO metadata and delivery via standard services may help them meet mandated data publication requirements. US Federal Data, for example, are supposed to be available via data.gov. ISO 19115 compliant metadata automatically generated from simulations and completed with standard web service endpoints can be harvested directly into data.gov and instantly become discoverable and viewable by the data.gov map viewer (Fig. 10).

Another benefit to the approach is that standardized datasets can be incorporated into an increasing number of applications, being developed around the world. The WMS services, for example, can be drag-and-dropped into applications like the Australian National Map (Fig. 11), which leverages the open-source terriaJS[14] library.

The standardized approach also has benefit to software developers: when you solve a problem to address a need in a specific application, you solve it for the whole community. For example, the CSW support we needed for the CMRE Matlab® users was added to NCTOOLBOX on Github, making CSW queries possible for any Matlab® user. The development of a common syntax for CSW bounding box requests in Matlab® helped not only Matlab® users, but all users of CSW, regardless of language.

Finally, a note on supporting tools in the environments scientists use: although most oceanographers still use Matlab®, Python is gaining in popularity, especially among younger researchers. Globally, Python is the fastest growing language over last 5 years[15] and is now the top teaching language at universities[16]. We therefore developed tools in both Matlab® and Python, so that users could query CSW or access data from OPeNDAP data with CF conventions effectively without needing to learn a new language. We believe this is critical for the adoption of the approach by scientists. This

[14] https://github.com/TerriaJS/terriajs.

[15] http://pypl.github.io/PYPL.html.

[16] http://www.pcworld.com/article/2451880/python-bumps-off-java-as-top-learning-language.html,

OSD
12, 2655–2682, 2015

should be extended in the future to R[17], another popular environment used in data science, and for future languages as they become popular in the community.

6 Conclusions

We have developed an approach using procedures and software tools that make it easy for small research groups to transform their heterogeneous collections of non-standard files into a standardized web services framework that allows interoperable data delivery, search, preview and access. This approach enables researchers to spend less time on data manipulation tasks, which allows more time for science. Users are able to query for datasets in Matlab® or Python, extract just the data they need from the discovered endpoints, and analyse the extracted data without model-specific code. The approach also allows data to be selectively distributed to the public, assists data publication requirements, allows data to be explored with a variety of new tools, and also to be connected to larger systems of standards-based data, such as IOOS, data.gov and GEOSS.

The use cases were particularly useful for gathering feedback on how the approach could be improved, but also for introducing a larger section of the community to a standards-based approach. At CMRE, it is hoped that the rotational scientists will further spread the approach when they return to their home countries.

Although the use cases here involved meteorological and oceanographic model data, the approach could be applied to any structured grid model data (e.g. surface and ground water modelling). Work is currently underway to extend this approach to work seamlessly with unstructured (e.g. triangle-based) grids and staggered grids (cf. the sci-wms project[18]).

[17] R project, http://www.r-project.org

[18] http://sci-wms.github.io/sci-wms.

OSD
12, 2655–2682, 2015

Acknowledgements. The authors acknowledge CMRE for the use of data collected during the REP14-MED trial described in Sect. 4.1. Thanks to John Caron and the Unidata Program Center for their development of the THREDDS Data Server, and to Tom Kralidis of the Meteorological Service of Canada for leading the development of the pycsw and OWSLib Python packages. Any use of trade, firm, or product names is for descriptive purposes only and does not imply endorsement by the US Government.

References

Bergamasco, A., Benetazzo, A., Carniel, S., Falcieri, F. M., Minuzzo, T., Signell, R. P., and Sclavo, M.: Knowledge discovery in large model datasets in the marine environment: the THREDDS Data Server example, Advances in Oceanography and Limnology, 3, 41–50, doi:10.1080/19475721.2012.669637, 2012.

Bernholdt, D., Bharathi, S., Brown, D., Chanchio, K., Chen, M., Chervenak, A., Cinquini, L., Drach, B., Foster, I., Fox, P., Garcia, J., Kesselman, C., Markel, R., Middleton, D., Nefedova, V., Pouchard, L., Shoshani, A., Sim, A., Strand, G., and Williams, D.: The Earth System Grid: supporting the next generation of climate modeling research, Proceedings of the IEEE, 93, 485–495, doi:10.1109/JPROC.2004.842745, 2005.

Blower, J. D., Gemmell, A. L., Griffiths, G. H., Haines, K., Santokhee, A., and Yang, X.: A Web Map Service implementation for the visualization of multidimensional gridded environmental data, Environ. Modell. Softw., 47, 218–224, doi:10.1016/j.envsoft.2013.04.002, 2013.

de La Beaujardière, J., Beegle-Krause, C., Bermudez, L., Hankin, S., Hazard, L., Howlett, E., Le, S., Proctor, R., Signell, R., Snowden, D., and Thomas, J.: Ocean and Coastal Data Management, Proceedings of OceanObs'09: Sustained Ocean Observations and Information for Society (Vol. 2), Venice, Italy, 21–25 September 2009, edited by: Hall, J., Harrison, D. E. and Stammer, D., ESA Publication WPP-306, doi:10.5270/OceanObs09.cwp.22, 2009.

Howlett, E., Snowden, D. P., Signell, R. P., Knee, K. R., and Wilson, D.: Data management update for the integrated ocean observing system (IOOS®), Oceans – St. John's 14–19 September 2014, 1–10, doi:10.1109/OCEANS.2014.7003284, 2014.

Signell, R.: Model data interoperability for the United States integrated ocean observing system (IOOS), Estuarine and Coastal Modeling, 221–238, doi:10.1061/41121(388)14, 2009.

OSD

12, 2655–2682, 2015

Signell, R. P. and Snowden, D. P.: Advances in a distributed approach for ocean model data interoperability, J. Mar. Sci. Eng., 2, 194–208, doi:10.3390/jmse2010194, 2014.

OSD
12, 2655–2682, 2015

```
SAVE_DIR="/usr/local/tomcat-thredds/webapps/ROOT/iso"

THREDDS_SERVERS = {
    "cmre-roms":    "http://scsrv26v:8080/thredds/roms.html",
    "cmre-glider":  "http://scsrv26v:8080/thredds/catalog/models/glider/catalog.html"
}

for subfolder, thredds_url in THREDDS_SERVERS.items():
    logger.info("Crawling %s (%s)" % (subfolder, thredds_url))
    crawler = Crawl(thredds_url, debug=True)
    isos = [(d.id, s.get("url")) for d in catalog.datasets
            for s in d.services if s.get("service").lower() == "iso"]
    filefolder = os.path.join(SAVE_DIR, subfolder)
    if not os.path.exists(filefolder):
        os.makedirs(filefolder)
    for iso in isos:
        try:
            filename = iso[0].replace("/", "_") + ".iso.xml"
            filepath = os.path.join(filefolder, filename)
            logger.info("Downloading/Saving %s" % filepath)
            urllib.urlretrieve(iso[1], filepath)
        except BaseException:
            logger.exception("Error!")
```

Figure 1. Snippet of Python code to crawl THREDDS catalogs. This code crawls two catalogs that contain forecast model runs and glider data. © North Atlantic Threaty Organization, all rights reserved. Provided by STO-CMRE (www.cmre.nato.int).

OSD
12, 2655–2682, 2015

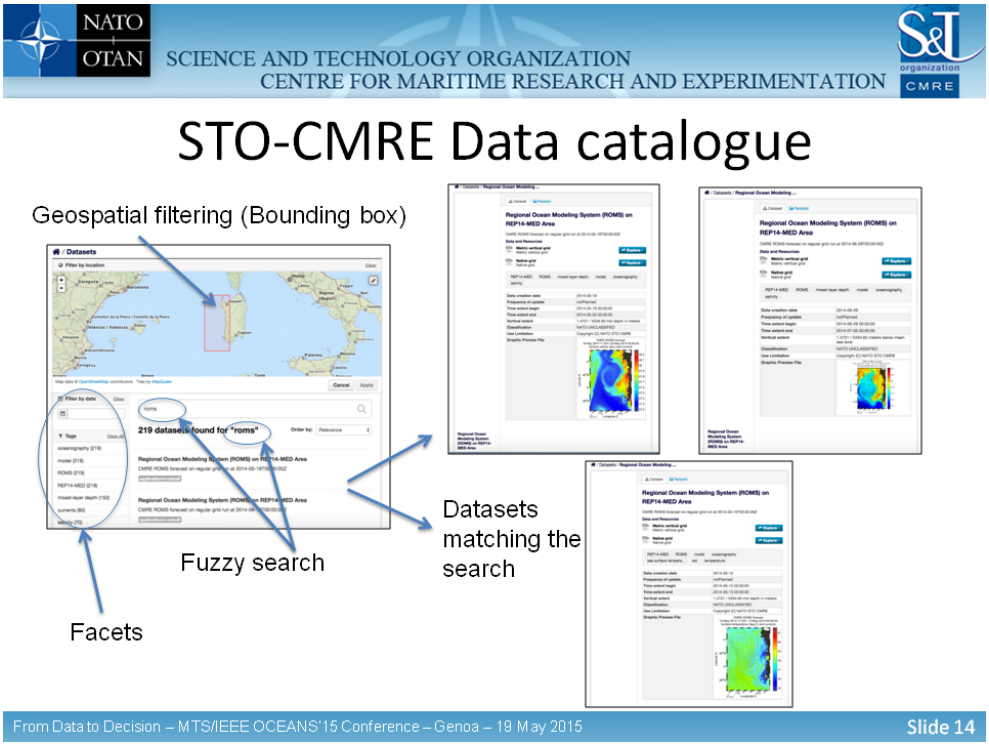

Figure 2. CKAN based interface and of the CMRE data catalogue. © North Atlantic Threaty Organization, all rights reserved. Provided by STO-CMRE.

OSD
12, 2655–2682, 2015

CC BY

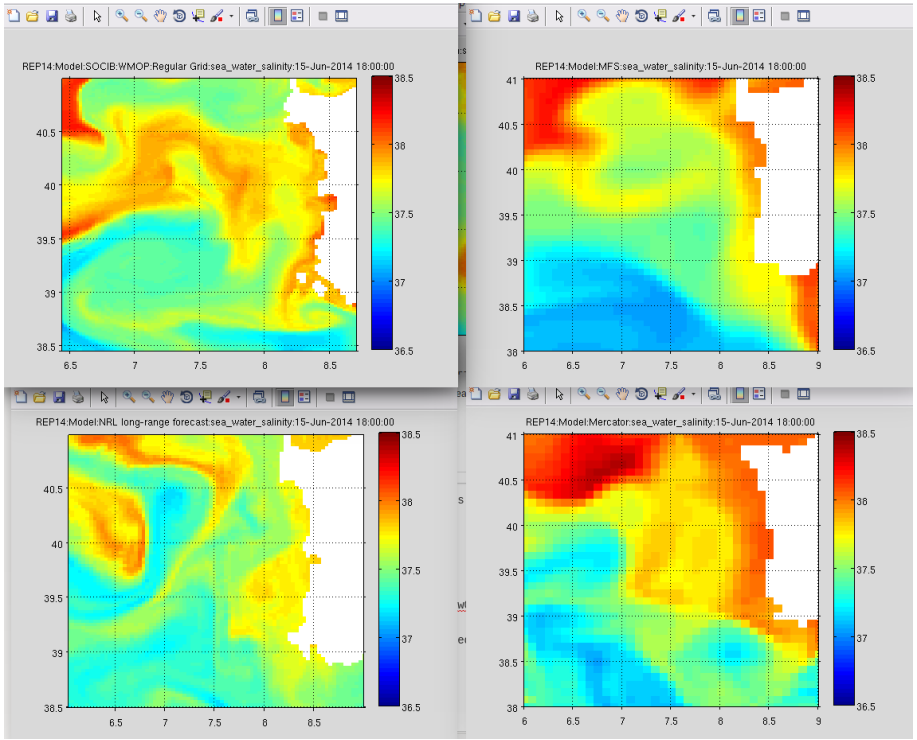

Figure 3. Comparison of surface salinity snapshots from four different models during REP14-MED Field Trial. By delivering data via web services with CF conventions, and using NCTOOLBOX which understands CF conventions, these figures were able to be made without any model-specific code. © North Atlantic Threaty Organization, all rights reserved. Provided by STO-CMRE.

OSD
12, 2655–2682, 2015

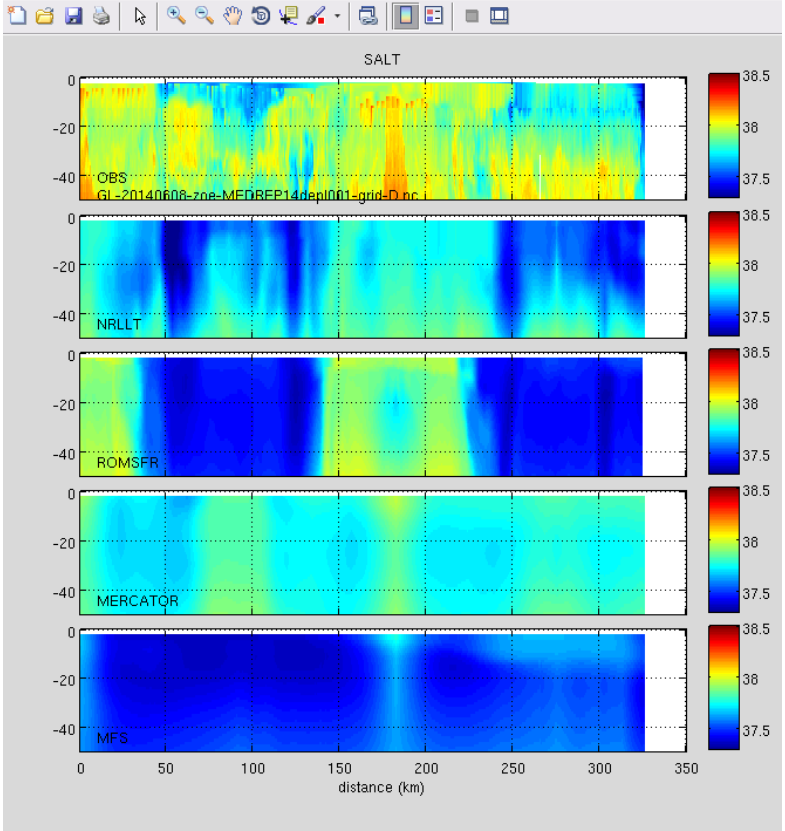

Figure 4. Vertical sections of salinity data from an ocean glider compared with virtual glider paths through four numerical models. Use of CF Conventions and NCTOOLBOX allowed the vertical coordinates from these four different models to be computed without any model-specific code. © North Atlantic Threaty Organization, all rights reserved. Provided by STO-CMRE.

OSD
12, 2655–2682, 2015

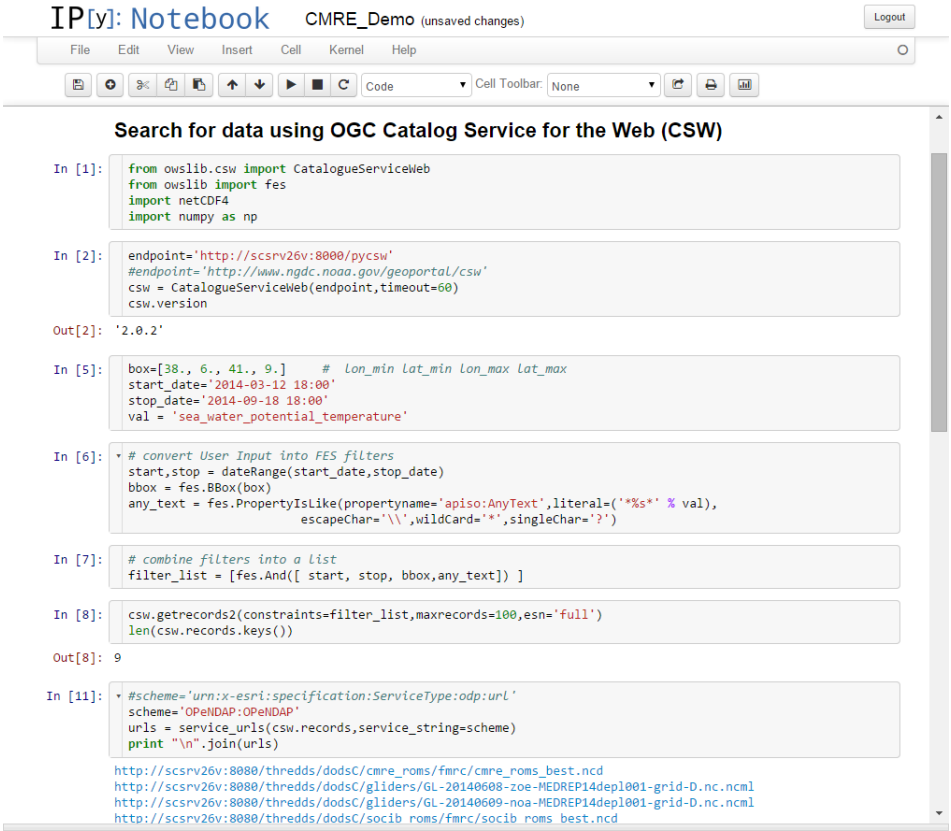

Figure 5. A snippet from an IPython Notebook demo, demonstrating a geospatial, temporal and free-text search for data using the owslib package to construct CSW queries and parse the results. Nine records are found, and the OPeNDAP data endpoints are then selected, from which data can be extracted in a common way, since CF Conventions are used. © North Atlantic Threaty Organization, all rights reserved. Provided by STO-CMRE.

OSD
12, 2655–2682, 2015

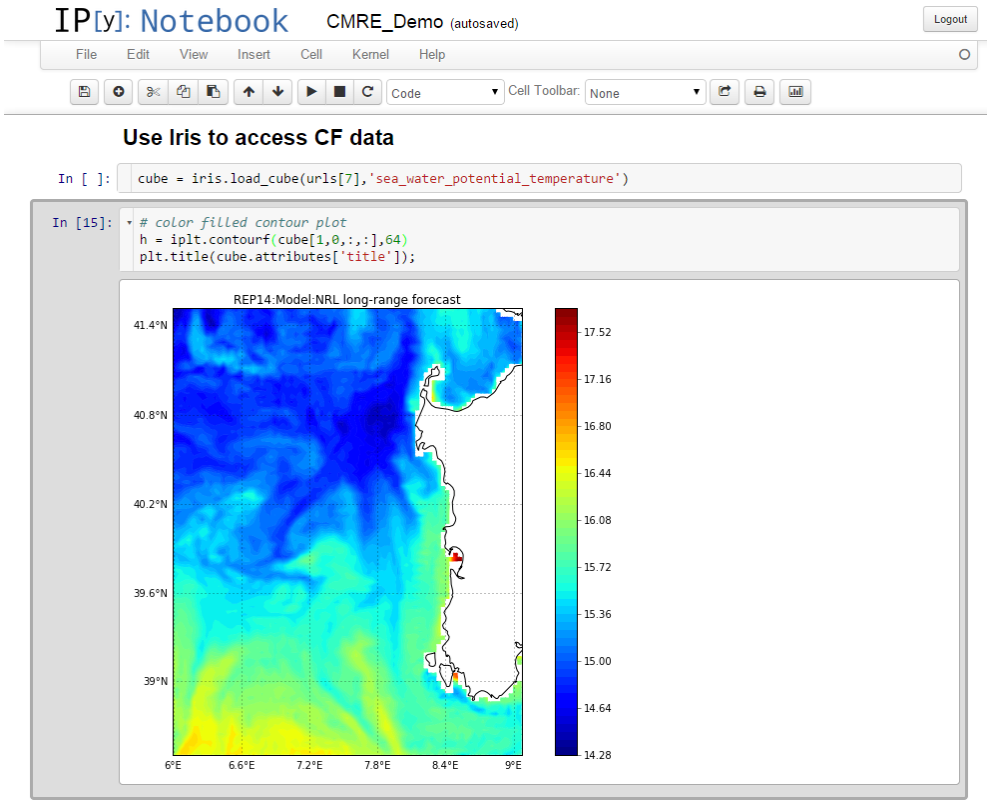

Figure 6. Another snippet from the Ipython Notebook demo, extracting data from one of the discovered OPeNDAP URLs using the Iris package. © North Atlantic Threaty Organization, all rights reserved. Provided by STO-CMRE.

OSD
12, 2655–2682, 2015

roms.yaml

```yaml
dataset:
    id: "USGS_COAWST_MVCO_CBLAST_Ripples_SWAN_40m"

    title: "USGS-CMG-COAWST Model: CBLAST2007 Ripples with SWAN-40m res"

    summary: "Simulation of hydrodynamics and bottom stress south of Marthas Vineyard, MA using

    project:
        - CMG_Portal
        - Sandy_Portal

    creator:
        email: nganju@usgs.gov
        name: Neil Ganju
        url: http://water.usgs.gov/fluxes

    publisher:
        email: tkalra@usgs.gov
        name: Tarandeep Kalra
        url: http://www.usgs.gov
```

Figure 7. Example of a YAML file, created by the modeller, and used to specify simulation specific parameters. This YAML file is converted into NcML using the yaml2roms python script. By specifying “CMG_Portal” in the Project section, the modeller is indicating that this dataset should be included in the USGS CMG_Portal web application.

OSD
12, 2655–2682, 2015

```xml
<?xml version="1.0" encoding="UTF-8"?>
<netcdf xmlns="http://www.unidata.ucar.edu/namespaces/netcdf/ncml-2.2">
  <attribute name="Conventions" type="String" value="CF-1.6, SGRID-0.1, ACDD-1.3"/>
  <attribute name="cdm_data_type" type="String" value="Grid"/>
  <attribute name="publisher_email" type="String" value="jcwarner@usgs.gov"/>
  <attribute name="publisher_url" type="String" value="http://woodshole.er.usgs.gov/sta
  <attribute name="publisher_name" type="String" value="John Warner"/>
  <attribute name="license" type="String" value="The data may be used and redistributec
  <attribute name="creator_email" type="String" value="jcwarner@usgs.gov"/>
  <attribute name="creator_url" type="String" value="http://woodshole.er.usgs.gov/staf
  <attribute name="creator_name" type="String" value="John Warner"/>
  <attribute name="title" type="String" value="USGS-CMG-COAWST Model: Hurricane Sandy,
  <attribute name="summary" type="String" value="Simulation of hydrodynamics, waves, bc
  <attribute name="project" type="String" value="CMG_Portal,Sandy_Portal"/>
  <attribute name="references" type="String" value="http://woodshole.er.usgs.gov/operat
  <attribute name="id" type="String" value="USGS_COAWST_Sandy_NYB05_sim6"/>
  <attribute name="naming_authority" type="String" value="gov.usgs.cmg"/>

  <variable name="bvstrcwmax">
  <attribute name="grid" type="String" value="grid"/>
  <attribute name="content_coverage_type" type="String" value="modelResult"/>
  <attribute name="location" type="String" value="face"/>
</variable>

<variable name="dep_net">
  <attribute name="grid" type="String" value="grid"/>
  <attribute name="content_coverage_type" type="String" value="modelResult"/>
  <attribute name="location" type="String" value="face"/>
</variable>

<variable name="v_stokes">
  <attribute name="grid" type="String" value="grid"/>
  <attribute name="content_coverage_type" type="String" value="modelResult"/>
  <attribute name="location" type="String" value="edge2"/>
</variable>

```

Figure 8. A sample NcML file produced by the yaml2ncml.py python script. Although these files can be created by hand using a text editor, getting validated NcML has proven difficult for modeller, and the yaml2ncml pathway has been found to be less error-prone. This NcML file not only specifies attributes, but also contains a regular expression that specifies which NetCDF files should be part of this aggregated dataset (not shown).

OSD
12, 2655–2682, 2015

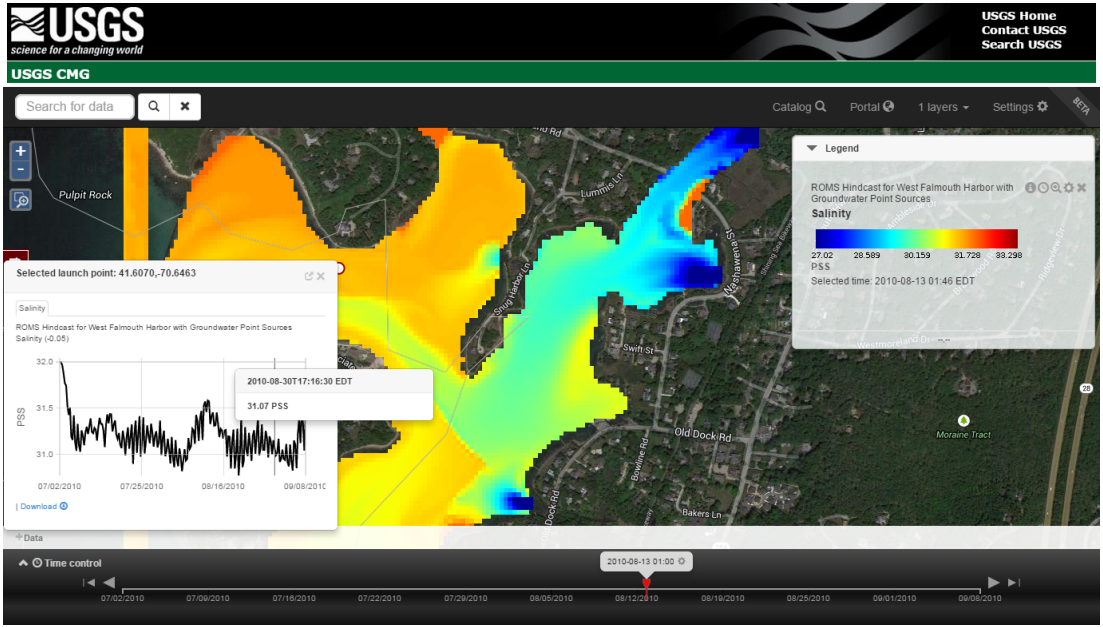

Figure 9. Viewing a USGS CMG dataset in the CMG_Portal web application. The map is made by a dynamic request to the WMS service provided by the THREDDS Data Server. The user has clicked a location on the map, which extracts a time series at that location using the WMS *getFeatureInfo* response.

OSD
12, 2655–2682, 2015

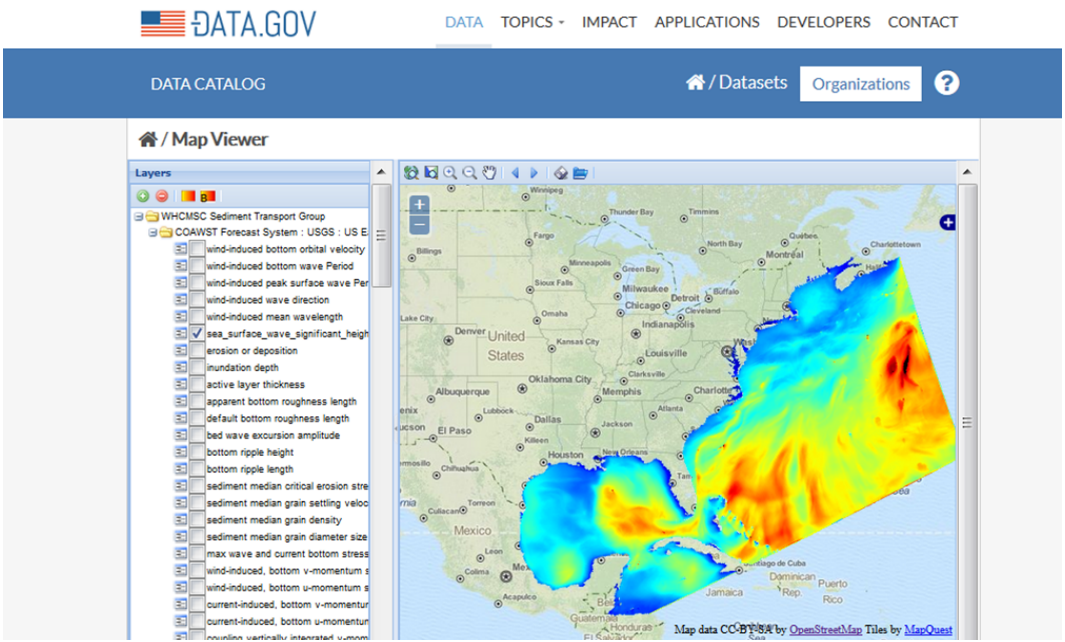

Figure 10. Viewing a discovered USGS CMG dataset in the Data.gov built-in Map Viewer application. The Map Viewer is requesting this image from the WMS service built into the THREDDS Data Server.

OSD
12, 2655–2682, 2015

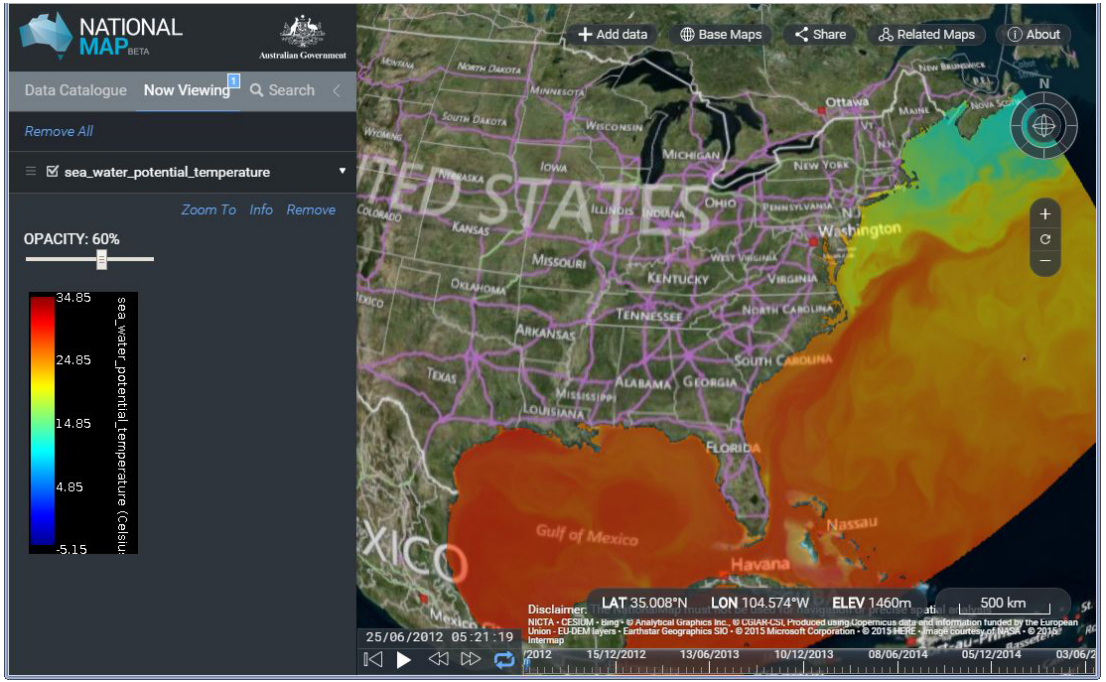

Figure 11. USGS CMG dataset displayed on the Australian National Map application. This map application is also requesting the image from the WMS service provided by the THREDDS Data Server.

OSD
12, 2655–2682, 2015