Manuscript prepared for Ocean Sci. Discuss.
with version 2014/09/16 7.15 Copernicus papers of the LaTeX class copernicus.cls.
Date: 21 March 2016

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 Science Technology Organization, Centre for Maritime Research and Experimentation, La Spezia, Italy

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

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 free technologies that are easy to deploy and configure; and (3) providing standardized, service-based tools that work in existing research environments. We present a simple, local brokering approach that lets modelers continue to use their existing files and tools, while serving virtual datasets that can be used with new standardized tools. The goal is to convince modelers that a standardized framework is not only useful, but can be implemented with modest effort using free software components. We use NetCDF Markup Language for data aggregation and standardization, the THREDDS Data Server for data delivery, pycsw for data search, NCTOOLBOX (MATLAB®[1]) and Iris (Python) for data access, and Open 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 global climate assessments to regional oceanographic studies. While systems such as Earth System Grid (Bernholdt et al., 2005) are in place for global climate projects, there are often no systems to search for and work with model data produced by smaller research groups. While these research groups may not produce the 10s of

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

Petabytes of data produced by the climate community, they often have significant holdings in the 10–100 TB range, too much to be delivered 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 conventions for specifying horizontal and vertical grid information, time units, variable attributes, requiring model specific tools.

Standardization allows the development of common tools that can access data from any model, removing the need for model-specific code. It also allows for cataloging of data, which allows researchers to easily retrieve simulations of interest by querying for specified geographic regions, time periods or keywords of interest. These benefits allow 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 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 facilitate data sharing at national, regional and global level, adopting a brokering approach that leaves resource providers free to use the model and data format they prefer, with data harmonized downstream of the service provider. This allows re-

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

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

searchers to continue using their existing custom data systems, while exploring the benefits of newer standardized data systems.

The same approach has been implemented by IOOS (Signell, 2009; Signell and Snowden, 2014) to achieve ocean data interoperability: IOOS 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.

Herein 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.

The main objectives of the paper are to (1) highlight the benefits of a standardized framework for ocean data, specifically in terms of facilitating scientists willing to share their data and enabling them to effectively access harmonized data (2) to demonstrate that a standardized framework can be implemented with modest effort integrating existing open source software components. For those seeking to implement the framework, full details on downloading, installing, configuring and using the components are maintained on the IOOS GitHub site[4].

We first describe the model data strategy used by IOOS, then describe the 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.

[4]https://github.com/ioos/model-data-framework/

2 The IOOS model data framework

The IOOS model data framework 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 Open-source Project for a Network Data Access Protocol (OPeNDAP) service with Climate and Forecast (CF) Conventions, and for images of data, via the OGC Web Map Service (WMS). OPeNDAP is used because Open Geospatial Consortium (OGC) services are not currently capable of delivering CF-compliant model data in a standardized way.

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[5] 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 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 (Fig. 1).

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 catalogue systems that can ingest metadata from ISO records or other sources, and enable searching

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

across distributed geospatial datasets, including GeoNetwork[6], Geoportal Server[7] , GI-cat[8], CKAN[9], and pycsw[10].

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 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[11], or even faster by using a Docker container[12]. 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

[6] http://geonetwork-opensource.org
[7] http://www.esri.com/software/arcgis/geoportal
[8] http://essi-lab.eu/do/view/GIcat
[9] http://ckan.org/
[10] http://pycsw.org
[11] http://www.unidata.ucar.edu/software/thredds/current/tds/tutorial/index.html
[12] https://hub.docker.com/r/axiom/docker-thredds/

in this directory, or in a subdirectory below this directory, are immediately accessible via the TDS.

The TDS can be installed on a server with modest computational resources, since the TDS doesn't actually do much work. It does, however, benefit from having a moderate amount of memory available (e.g. more than 4GB). In addition, because data requests to the TDS extract data in disk, users of the TDS will benefit from faster disk access (e.g. local disk over remotely-mounted disk).

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 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. If the dataset has one dimensional longitude, latitude and depth coorindates, it can also be delivered via the OGC Web Coverage Service (WCS). This NcML approach is convenient for model data providers as no reload or restart of the TDS is necessary. This is quite useful, as modelers often have a large collection of simulations that test the sensitivity to changes in parameterizations, data assimilation techniques, boundary conditions, etc.

For forecast models, a slightly different approach is needed, as forecast time periods overlap and the files cannot be simply joined along the time dimension. In this case, the NcML Forecast Model Run Collection is used, which virtually creates a "best time series" and other dataset views of the forecast collection. This type of NcML must be added to static TDS catalogues. While new forecast data can be added to forecast aggregation without reloading the server, any modification to the static catalogues 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. This allows them to extract just the part of the dataset they need directly into their workflow, without downloading any files. 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. Although we focus on MATLAB and Python here, any software that can read NetCDF files can open the URL of a remote OPeNDAP dataset just as if it were a local NetCDF File. There is a list of over 50 software packages that can read NetCDF and OPeNDAP maintained by Unidata, including ArcGIS, IDL, Mathematica and more[13].

For MATLAB®, NCTOOLBOX[14] provides support for the CF Conventions by leveraging the Unidata NetCDF-Java library behind a simple MATLAB® interface. For Python, the Iris package[15] from the British Met Office provides support for the CF Conventions by leveraging the Unidata NetCDF C library behind a Python interface. Both of these tools allow users to work with any CF-Compliant model without writing any model-specific code: the user requests data from a specified time and geospatial extent, and the software automatically returns the longitude, latitude, depth and time coordinates in standard form, regardless of the variable naming conventions, time units, and vertical coordinate system specification in the original native model data files. This greatly facilitates model-model comparison and skill assessment, as users can bring in data from a variety of models without specialized knowledge of any of them.

[13]Software for Manipulating or Displaying NetCDF Data, available at http://www.unidata.ucar.edu/software/netcdf/software.html

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

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

3.3 Data Search

Although there are many free metadata catalogue systems that can enable geospatial and temporal searches across distributed data, here we use pycsw, a lightweight system that provides an OGC-certified compliant catalogue Service for the Web (CSW) search capability. It is easy to install, configure and maintain via the command line, allowing easy scripting capability. The CSW service allows for complex queries to be constructed. These queries often include a geospatial bounding box, a temporal extent, and specific text strings that should be either included or excluded. We populate the pycsw catalogue by ingesting ISO 19115 compliant metadata records that have been collected from selected TDS datasets.

The TDS datasets selected by running a Python script that crawls a list of local or remote THREDDS catalogues, enabling filtering on specific types of datasets (see the code in Fig. 2). For example, we can specify for a specific THREDDS catalogue that only metadata from .ncml files should be harvested, while 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 WMS services included in the THREDDS Data Server are provided by the ncWMS package (Blower et al., 2013), which introduced some important extensions[16] to effectively create maps from large multidimensional arrays of data. The most important of these extensions are the ability to specify color scale ranges. The OGC originally developed WMS to deliver static images in a variety of projections in response to queries that specified a specific bounding box and time. The TDS ncWMS extends this to deliver dynamically-generated images that map data onto color ranges specified by the user. This allows users,

[16] http://www.resc.rdg.ac.uk/trac/ncWMS/wiki/WmsExtensions

for example, to effectively explore structure from global scale to local bays, and from winter to summer conditions.

There are many clients that can consume WMS services, including GIS packages such as ArcGIS and QGIS, but these don't typically provide support for the custom extensions provided by ncWMS. Fortunately, the TDS also includes the Godiva2 WMS client which allows for simple preview of individual datasets by providing a web-based GUI that interfaces with the ncWMS services (Blower et al., 2009).

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. CMRE regularly conducts focused oceanographic cruises in various regions of the ocean, involving multiple Nations and research institutions, who gather to run experiments integrating data collected from heterogeneous ocean observing networks comprising variegate assets, such as in situ sensors, remote sensing and numerical models.

During oceanographic experiments, real-time data are ingested in the CMRE scientific data management infrastructure, facilitating data sharing among CMRE scientists and partners. A continuous deployment service automatically extracts metadata from datasets and populate the CMRE data repository and data catalogue, and newly acquired observations and data products become available to CMRE researchers in near real-time through the catalogue interface. The CMRE data management infrastructure adopts open standards for data interoperability and OGC services for geospatial data, enforcing compliance to NATO

(Allied Geographical Publications, STANAGS) and international standards for spatial data (ISO 19100 family, ISO/TC 211).

The first version of the data management infrastructure was heavily tailored on the data formats and the best practice adopted for data within CMRE. It integrated Python scripts, a web-based scheduler (Jenkins), and versioning software (Git) to extract metadata from datasets and populate the data repository, which combined a file storage and a data server (GeoServer). The GeoServer was used to implement OGC WCS and WMS services, while CKAN and GeoNetwork were used to provide a user-friendly interface for data search.

This version of the data management infrastructure, leveraging OGC services, worked well for observed data, but services were not in place to access model data. Because both observed and model data are required for model skill assessment, modelers were not using the existing infrastructure for these activities. They stored NetCDF files on local hard drives and for visualization and analysis, ran model-specific routines for each type of model, mainly written in MATLAB®. Moreover, exchange of data with other scientists was mainly done via FTP or via exchange of physical media. As a result, duplication of files and loss of data were frequent.

To advance the management of ocean data, a prototype demonstrating the benefits of the IOOS approach was developed and tested. Data from REP14-MED cruise, conducted in 2014 in the Sardinian Sea, were used for evaluating the approach. 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. Knowledge on standard conventions is required to create the initial NcML file, but then it can be hand off to modelers to copy, modify and use on their own. The same approach was used also for observations, in particular to enforce CF-compliance for glider data.

In both cases, no modification of the original data was necessary, because the NcML transformations were applied on the fly when accessing the data through OPeNDAP. Since

the original data generation workflow was not affected in any way, legacy software can continue to access the data using the original formats. However, the integration in the data management infrastructure of the OPeNDAP service empowered the scientists with a powerful alternative for accessing data, letting them free to access only the subset of data they really need, avoiding to download huge files on they local hard drives. Moreover, to share the data it is now sufficient to share the URL pointing to the dataset, avoiding unnecessary duplications and use of external media. This is a key advantage of the approach, in particular when, as in this case, such data are used as an input source of information by several users and services.

To test the potential integration with the CMRE data catalogue, the ncISO service was used to generate automatically metadata directly from the datasets ingested in THREDDS. To crawl the specific THREDDS catalogue created for REP-14-MED data, we deployed the Python script in Fig. 2. The generated metadata are ISO 19115 compliant, and can be published through a OGC CSW server such as pycsw and be harvested by geo-catalogue, such as CKAN or GeoNetwork. Compared with the original approach used for metadata extraction, ncISO metadata generated from THREDDS have the advantage to describe better some dataset features, such as as variable ranges. Other features, such as cruise information, can be customized through NcML and the Extensible Stylesheet Language schema which is used for configuring ncISO outputs.

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 search capability for MATLAB® users. Python users had the powerful OWSLib[17] library, but nothing existed for MATLAB®. To meet this need, we developed a simple CSW search 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 demonstrate the approach, we first extracted sea surface salinity from all REP14-MED models in the CMRE data catalogue. We accomplished this by formulating a CSW search for the REP14-MED region and time period in MATLAB®, including a search for the text

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

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 compared interpolated glider data to extracted profiles from four models along the 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[18], 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 installation and configuration of the system took place while the first author was visiting CMRE for one month under the CMRE Visiting Researcher Programme. The prototype was integrated in the production data management environment during another cruise in 2015. During that cruise, model data and glider observations where ingested automatically in THREDDS, which currently integrates the data repository, by the continuous deployment service. The same service that upload the datasets in THREDDS takes also care of generating customized NcML files and xsl descriptions for generating enriched ISO metadata using the ncISO service. These metadata includes customized descriptions of datasets as defined by data producers, but contain more precised information on the specific datasets as presented by THREDDS.

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

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-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. In addition, there is a lot of repetitive attribute information that needs to be added for each variable, which varies from simulation to simulation.

To solve this problem, we simplified the editing task by having modelers edit a simple YAML[19] 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. It is not specific to USGS, and could be used to simply the generation of NcML at CMRE or within any modeling group.

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 ISO metadata, and becomes queryable via the CSW service. An example of the YAML input file and the

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

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). This means that the USGS modelers control which simulations appear in the portal without any involvement of the portal developers.

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 smaller research groups, the automatic generation of ISO metadata and delivery via standard ser-

vices 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[20] 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[21] and is now the top teaching language at universities[22]. 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 should be extended in the future to R[23], another popular environment used in data science, and for future languages as they become popular in the community.

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

[21] http://pypl.GitHub.io/PYPL.html

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

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

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 framework 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. This greatly facilitates model-model and model-data comparison. Because services are used to extract data, users can execute their analysis and visualization workflows not only on their computers in the office, but using their laptop in a hotel room, or anywhere. This also means workflows shared with colleagues will work without modification. Although reading data with services is slower than reading data from disk, obtaining just the data you need via a service is faster than downloading an entire dataset and then reading the file locally. If data obtained via services needs to be read repeatedly, it can be saved to local disk for faster subsequent access.

The framework 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 framework could be applied to any structured grid model data (e.g. surface and ground

water modelling). Work is currently underway to extend this framework to work seamlessly with unstructured (e.g. triangle-based) grids and staggered grids (cf. the sci-wms project[24]).

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.

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

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.

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.

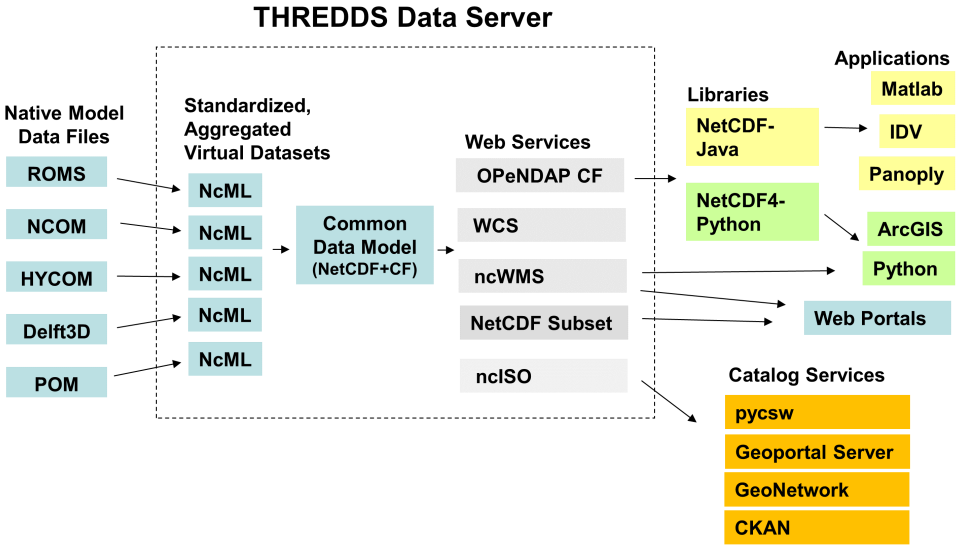

Figure 1. The IOOS Model Data Framework. Collections of non-standard NetCDF (or GRIB) files are transformed via NetCDF Markup Language (NcML) into aggregated, standardized datasets that can be represented internally in the THREDDS Data Server in a common data model. This data in common form can then be delivered via a number of standard services, including map services (WMS), data services (OPeNDAP, WCS and NetCDF Subset Service), and metadata services (ncISO). This services in turn feed web portals, scientific analysis workflows in Python or MATLAB, and catalogue services.

```
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 2. Snippet of Python code to crawl THREDDS catalogues. This code crawls two catalogues that contain forecast model runs and glider data. Provided by STO-CMRE (www.cmre.nato.int).

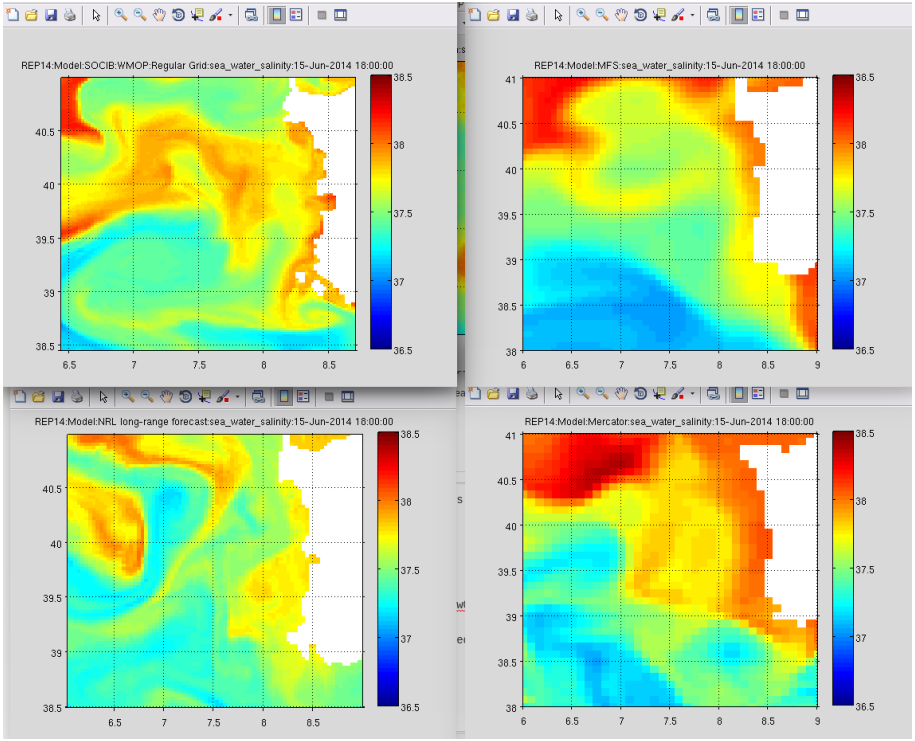

Figure 3. Comparison of surface salinity snapshots from four different models during REP14-MED Field Trial. By delivering data via web services with CF conventions, and using NCTOOLBOX which understands CF conventions, these figures were able to be made without any model-specific code.

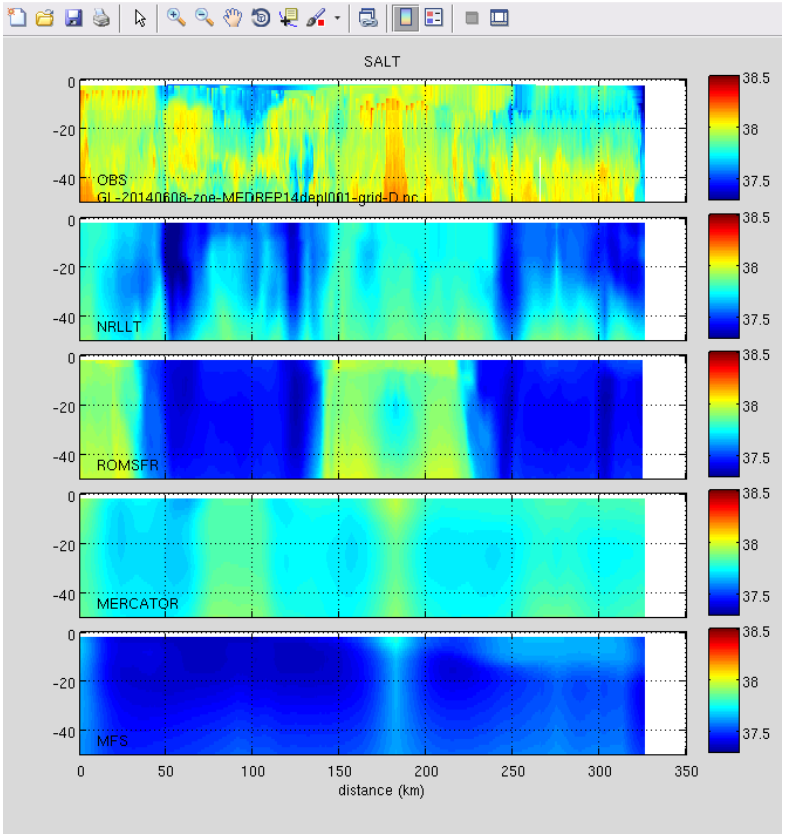

Figure 4. Vertical sections of salinity data from an ocean glider compared with virtual glider paths through four numerical models. Use of CF Conventions and NCTOOLBOX allowed the vertical coordinates from these four different models to be computed without any model-specific code. © North Atlantic Threaty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

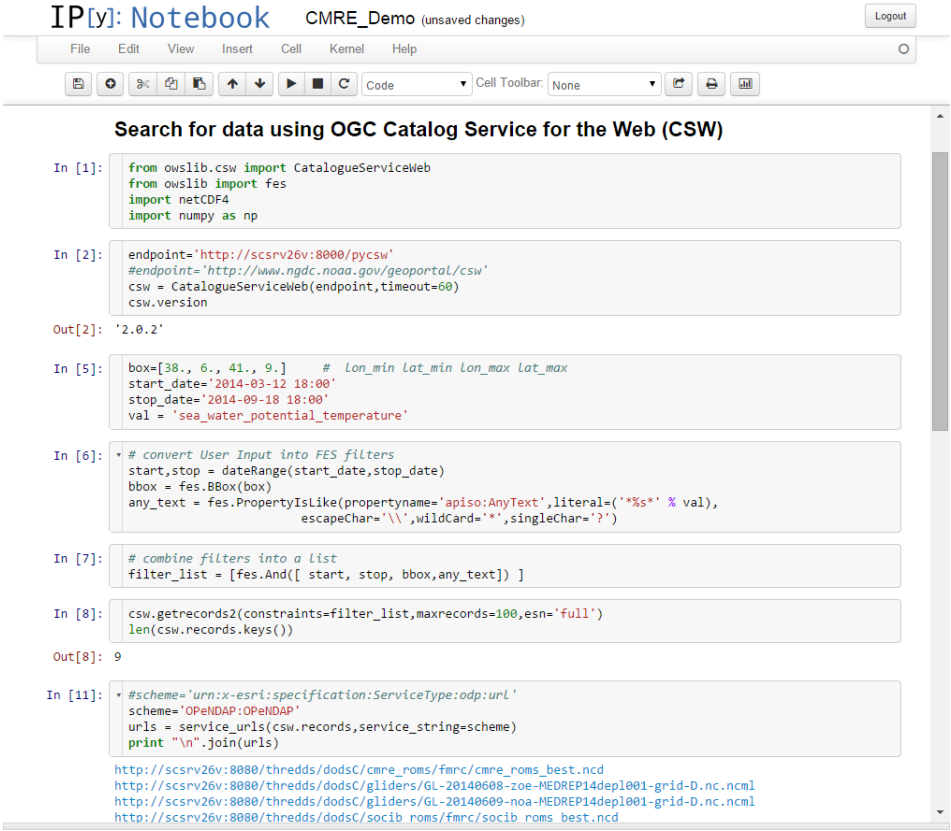

Figure 5. A snippet from an IPython Notebook demo, demonstrating a geospatial, temporal and free-text search for data using the owslib package to construct CSW queries and parse the results. Nine records are found, and the OPeNDAP data endpoints are then selected, from which data can be extracted in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

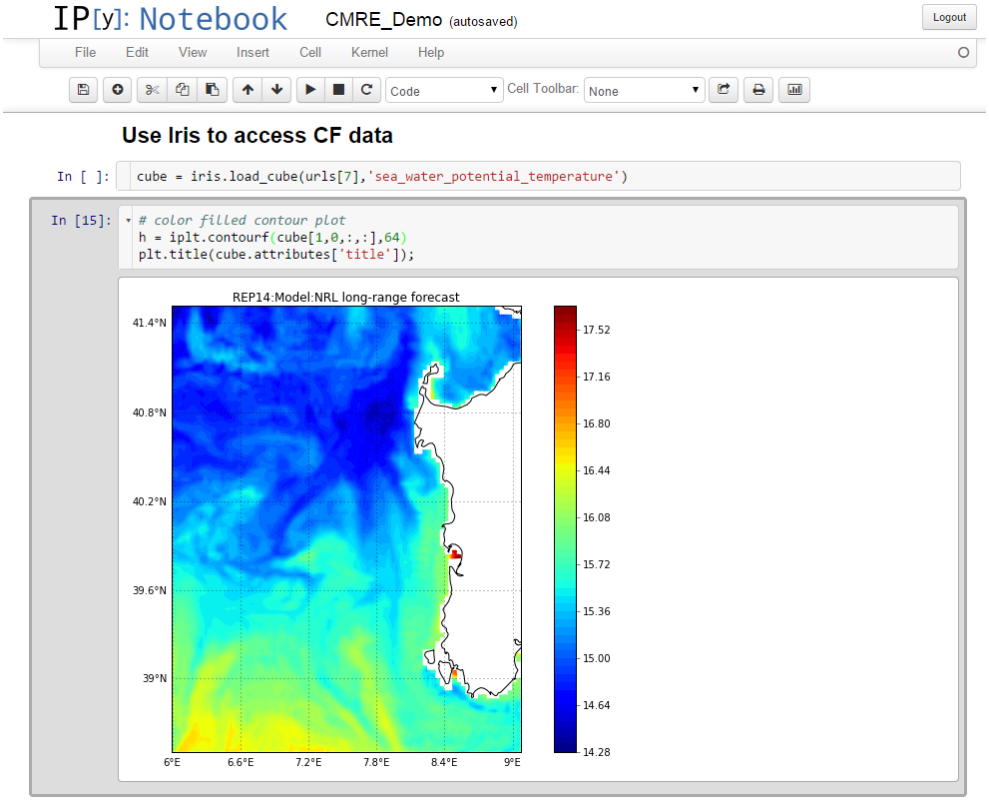

Figure 6. Another snippet from the Ipython Notebook demo, extracting data from one of the discovered OPeNDAP URLs using the Iris package. © North Atlantic Threaty Organization, all rights reserved. Provided by STO-CMRE.

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.

```
<?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 redistributed
  <attribute name="creator_email" type="String" value="jcwarner@usgs.gov"/>
  <attribute name="creator_url" type="String" value="http://woodshole.er.usgs.gov/staff
  <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, bo
  <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).

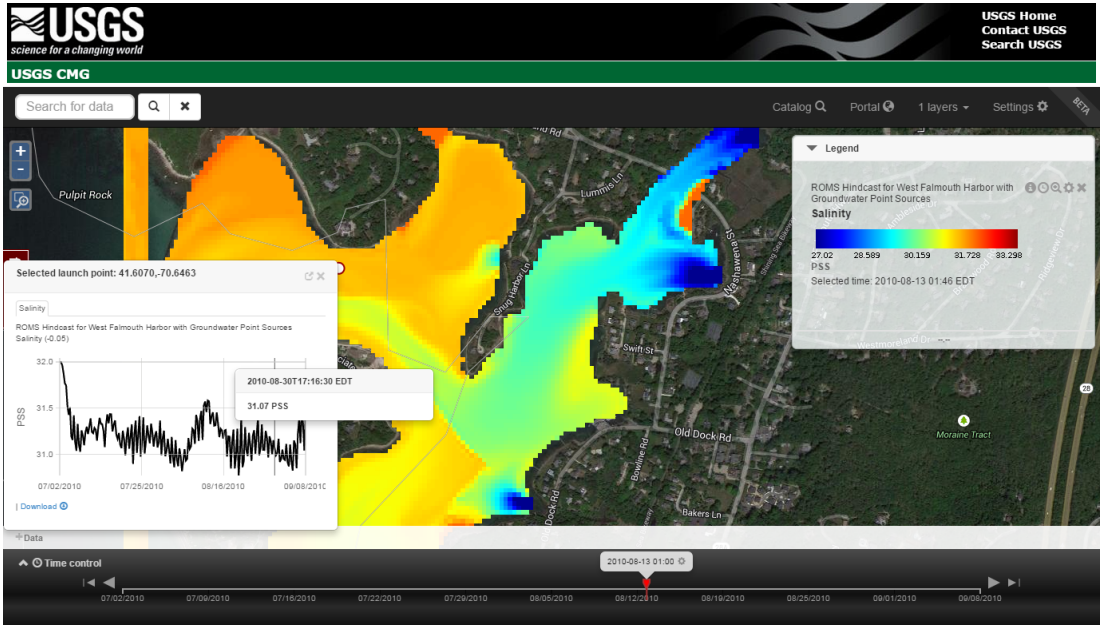

Figure 9. Viewing a USGS CMG dataset in the CMG_Portal web application. The map is made by a dynamic request to the WMS service provided by the THREDDS Data Server. The user has clicked a location on the map, which extracts a time series at that location using the WMS *getFeatureInfo* response.

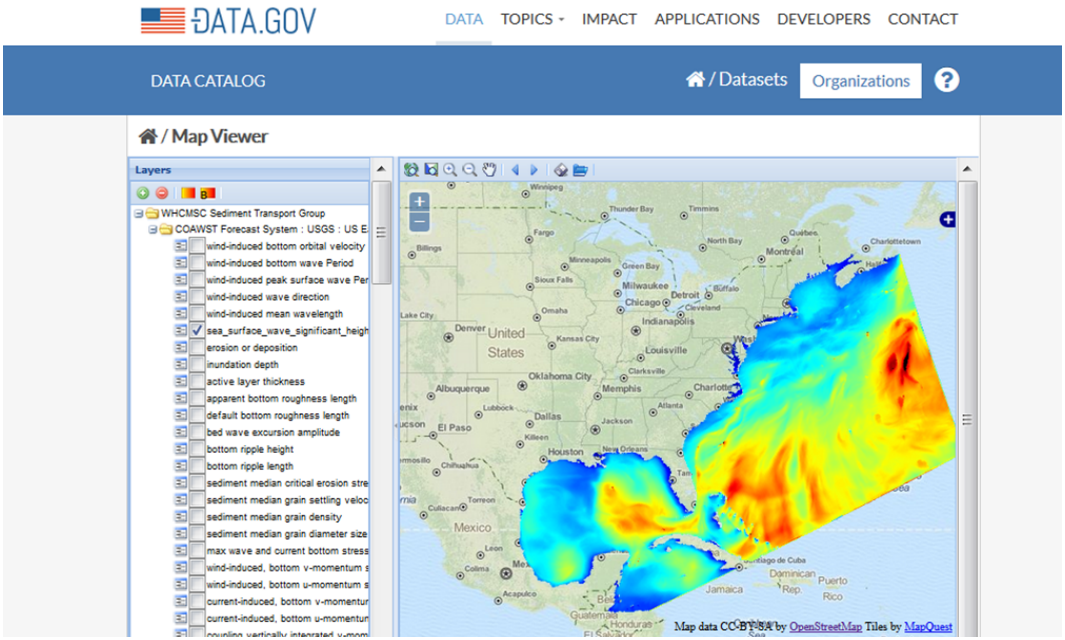

Figure 10. Viewing a discovered USGS CMG dataset in the Data.gov built-in Map Viewer application. The Map Viewer is requesting this image from the WMS service built into the THREDDS Data Server.

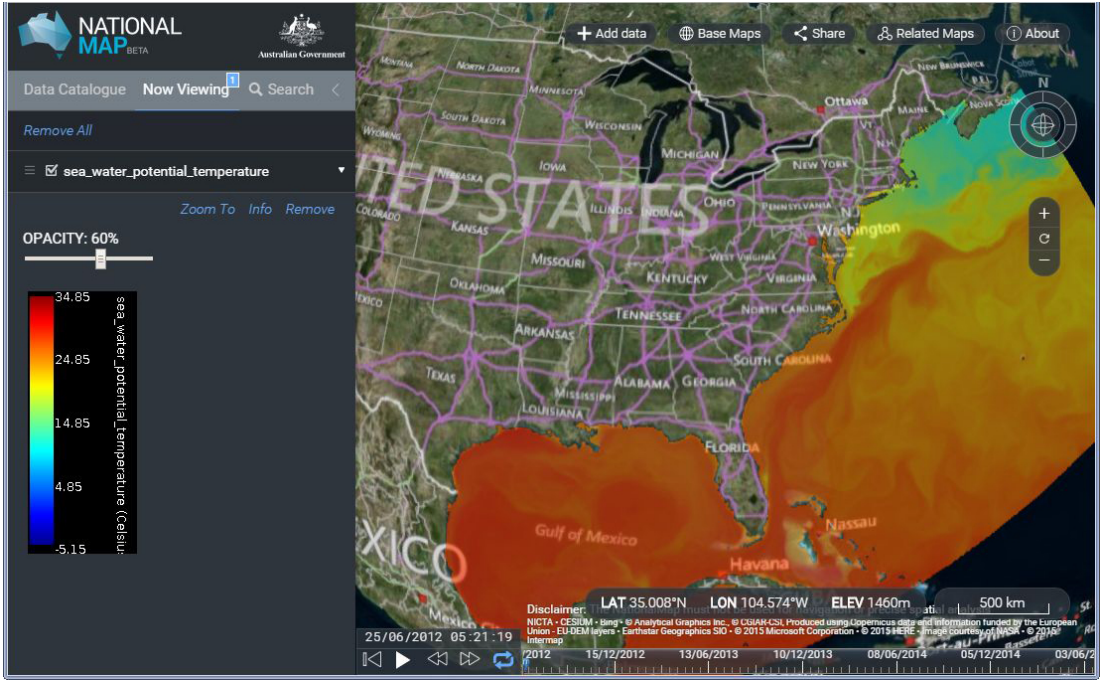

Figure 11. USGS CMG dataset displayed on the Australian National Map application. This map application is also requesting the image from the WMS service provided by the THREDDS Data Server.