Ocean Sci. Discuss., 12, C1511–C1518, 2016
www.ocean-sci-discuss.net/12/C1511/2016/

Ocean Science
Discussions

Interactive comment on "Technical note: Harmonizing met-ocean model data via standard web services within small research groups" *by* R. P. Signell and E. Camossi

R. P. Signell and E. Camossi

rsignell@usgs.gov

Received and published: 26 January 2016

Reviewer comment: This manuscript describes how data services can be implemented at a local level to help with data management and distribution within individual research groups. The paper promotes practices that are rapidly becoming "standard" but thus far are typically deployed at larger data/modeling centers.

While I applaud this goal, I think the manuscript needs some work before it can be published. I think at some level the paper lacks focus. It's not clear to me what the main theme is, e.g., whether it is advocating smaller modeling groups to setup THREDDS servers, or it is attempting to show how straightforward it is, or is it trying to

show what the benefits (by way of example) are in doing so?

I suspect the authors are trying for all three, but the result is none are fully explored. It certainly would not be possible to set up such a system solely based on this manuscript. Likewise, it's not quite clear what such a system would look like or concrete examples of how it works. Further, there appears to be a mix between fine detail, e.g., the use of datasetScan, and more big-picture types of descriptions that are not fully explained, e.g., "use pycsw". I would recommend the authors try to key on one issue and fully develop that. It should be noted, however, that such a paper could become more like a "how to", or user guide. Instead, I think showing concrete examples, perhaps case studies with and without data services, would help.

Reply: Indeed, this paper has two goals, to convince small groups that: (1) a standardized framework is useful, and (2) a standardized framework can be implemented with modest effort using free software components. The blend of big-picture and detail was actually intentional. If we stayed completely at the big-picture level, the paper would be too generic and likely would convince nobody that this approach was worth taking. As the reviewer suggests, if we focused only on one aspect of the framework and provided full detail the paper would be more of a technical report or cookbook and again, would interest few readers. We hope to convince scientists to become interested in implementing the framework, and if interested, they will find more detailed information which is readily available on the web. We have modified the introduction to make it more clear that this is the intent, and have added web site reference for framework implementation details.

Some more specific points:

1. The paper uses a lot of acronyms, out of necessity, but only occasionally (and almost randomly) are these spelled out. In some cases it might be obvious, e.g., NATO. On the other hand, it's probably good practice to either spell them all out on

first use or include a list of acronyms at the end. For example the asbtract alone has netCDF, THREDDS, pycsw, NATO and USGS; OGC on the other hand is spelled out four different times.

Yes, there are a lot of acronyms, but it was not random which ones we chose to spell out. In some situations, like "pycsw", it's not actually an acronym, but a package name. And in other situations, like "THREDDS", the acronym spelled out (Thematic Real-time Environmental Distributed Data Services) does not give additional insight, and in fact is never used in practice. Thus it's just a name. Having the OGC spelled out four times, of course, is in fact an error. These and other multiple acronym expansions have been fixed.

2. It might be a minor, or sensitive point, but I prefer the phrase "model output" over "model data". If nothing else, this would help with potentially complex phrases that include "model data" and "data model", such as lines 24-25 "model data infrastructure ... for data models". This is just a suggestion, not a criticism.

We understand the point, but we prefer "model data" over "model output", because "model output" could be misconstrued to mean plots or other types of visualizations or derived products. We use "model data" because we are referring to the multidimensional arrays of geospatial data that just happen to be generated by a simulation rather than by a measured by a particular sensor. Thus we prefer the terms "model data", "in situ sensor data", "satellite derived data" to denote the different origins. It's true that "model data" and "data model" are very different concepts, but both are commonly used in the field (e.g. https://www.ncdc.noaa.gov/data-access/model-data, https://en.wikipedia.org/wiki/Data$_{model}$)

3. The authors don't give any measurable quantification of using a data service instead of a regular file system. Is there a way to produce usage metrics, for example, with THREDDS that can't otherwise be done (thus giving the data providers a better

idea of who is using the output)? Or metrics on access speeds? In other words, if I have a small modeling group and don't really care about data discovery issues, why would I go through the trouble? it is faster? Can I more easily track users?

The benefits for a researcher are more qualitative than quantitative, but they are numerous: to be able to search for model data within their group, to create aggregated virtual datasets from piles of files (but without breaking existing tools or workflows), to automatically generate metadata that is now often required by sponsors, to use tools that don't require model-specific code, and benefit from an ever increasing set of tools for standardized data. There are likely some situations where access speed is more important than these benefits and custom non-standard software is required, but these special cases are the exceptions, not the norm. We will modify the text to more clearly summarize these benefits.

4. THREDDS has a limitation that input data must be in netCDF format. While netCDF is certainly a standard, what happens if modeling groups produce output in multiple grib and/or flat binary? Would they have to setup another OPENDAP server? THREDDS actually handles grib files natively as well, and the software uses a plug-in architecture that allows providers to write a custom I/O Service Provider module if they want to keep their existing archives of custom binary data. We will modify the text to reflect this.

5. As far as I know, THREDDS will require an apache tomcat server. It's not clear what sort of requirements this puts on the server machine. For example, groups may not want to overload a production machine (running models) with a data service that could overwhelm the machine resources. Or, perhaps this is not a drain on the machine memory and/or CPU?

THREDDS is very lightweight in terms of CPU requirements, but does function better on machines with lots of memory (e.g. 16 or 32GB). We will add some

information on server requirements to the text.

6. The abstract describes data services in a somewhat independent way, e.g., pycsw is used for data discovery, THREDDS for data delivery, etc. However, the situation is more parallel (I think). It all starts by having netCDF data. TDS and NCML then expose these data via OPENDAP to client tools. At the same time, TDS creates ISO metadata records that can be harvested by pycsw. And, TDS can be configured with a builtin tool providing data browsing capabilities (WMS).

Yes, this is exactly right. We will add a figure which indicates how the data is transformed from non-standard NetCDF files to standardized data and metadata services within the THREDDS Data Server and then flow into applications such as Matlab and Python, and feed catalog systems such as pycsw, something like slide 7 here (https://speakerdeck.com/rsignell/catalog-driven-workflows-using-csw).

7. Any comment on the advantage of pycsw over the other CSW mentioned on page 4 (lines 15-18)?

We will simplify the discussion of the different search APIs, and make it clear that with CSW, the user can construct more sophisticated queries than OpenSearch or CKAN. We will add text that clarifies that pycsw is not necessarily better than other CSW solutions, but it is simple to install, maintain and configure.

8. Section 3.1, machine resources needed for TDS? Want to add ability of NCML to "modify" output, e.g., hide variables, rename, add metadata, etc.?

We will add a description of the required machine resources for TDS. We will also add text about the power of NcML to virtually modify the metadata.

9. Section 3.2 might be cut. It's too short to be meaningful; maybe add discussion into 3.1, e.g., "opendap enabled tools".

We can expand Section 3.2 a bit. It is important because here we specify the Matlab and Python tools that take advantage of the standardized framework.

10. Ditto section 3.4 (WMS not fully explained).

We will add text regarding these WMS services that indicate the extensions that are required for effective access to model output (such as color range, log vs. linear scale).

11. Include figure for section 3 that shows integration of these? Maybe of Godiva2?

The integration of these is shown in section 4, the Use Cases section.

12. I'm not sure I understand the bottom paragraph on page 7; "During the trial"? Using GeoServer not TDS? CKAN not pycsw?

We will remove the description of the NATO system used prior to implementing the framework described here, as it distracts from the main points of this paper.

13. The example display of glider output is interesting, but not really in line with the main theme of model output and data services. In addition, it opens a lot of questions, for example, how the lat/lon/depth are interpolated and deconvolved with time. I think with such color-shaded plots it is assumed that the glider up/down is done instantly in time? Otherwise is it better to display these as saw-tooth tracks (up/down)?

The purpose of this plot was to demonstrate that the framework can allow comparison of data from models with different vertical coordinate systems, without any model-specific code. We will add text that explains how the glider data were interpolated onto a vertical section here because the up/down length of each glider segment is short compared to the scales being compared here.

14. Section 4 introduces Ipython notebook, which is very interesting, but somewhat outside the scope of the rest of the paper.

We want to include the notebook as a powerful example of the entire user

workflow, from catalog search, to data access, to data analysis and display.

15. Section 4.2 is also somewhat cursory; it gives very specific details about the TDS implementation at USGS. Could this be re-written to include a more specific account of a) what was needed; b) how it was implemented; and c) what the benefits are?

Agreed. This section has been expanded to reflect the experiences and rationale behind adding these scripts to make it even easier for scientists to generate standardized aggregations.

16. In Discussion, mention other benefits, such as proper cataloging of model runs, exposure to other TDS catalogs, "standarization" of output, etc.

Agreed. This section has also been expanded to mention these additional benefits of using a standardized framework.

Interactive comment on Ocean Sci. Discuss., 12, 2655, 2015.

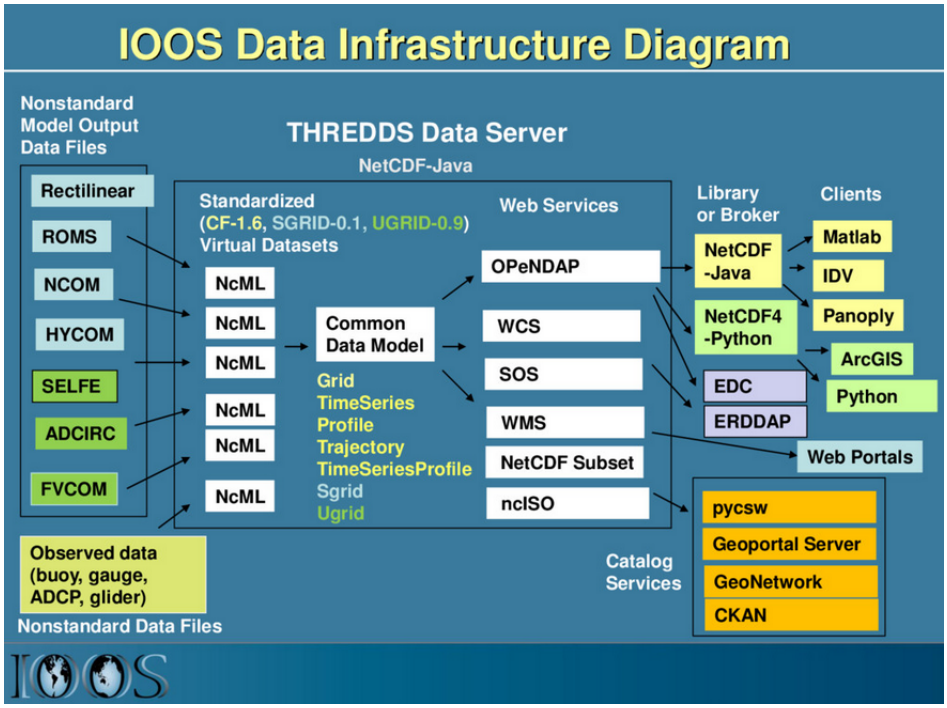

Fig. 1. Sample figure to replace current figure 2.