A Unified Framework for IOOS Model Data Access

Proposal Submitted in Response to NOAA Federal Funding Opportunity
#NOAA-NOS-IOOS-2018-2005452
UCAR Proposal # 2018-0196

Funding Type: Cooperative Agreement
Applicant Type: Non-profit

Principal Investigator
Sean C. Arms
Software Engineer, UCAR/Unidata

Institutional Representative
Amy Smith
Manager, UCAR Contracts – Direct Awards

Table of Contents

Project Summary

Project Title

A Unified Framework for IOOS Model Data Access

Primary Contact

Scientific Principal Investigator
Sean C. Arms
Software Engineer, UCAR/Unidata
[redacted]

Primary Recipient Institution

UCAR
Amy Smith
Manager, UCAR Contracts - Direct Awards
[redacted]

Investigators

Transition Principal Investigator
Kyle Wilcox
Software Architect, Axiom Data Science
[redacted]

Brief Summary

We propose to create a unified framework for IOOS model data, extending Unidata's NetCDF-Java and THREDDS Data Server to fully support the unstructured grid models now being widely used in the IOOS community. This will fulfill a need that has existed for more than a decade, and will produce services and tools that have direct, immediate benefit to the operational IOOS Federal Backbone community as well as IOOS Regional Association partners.

Partners

- NERACOOS
- GLERL
- NOAA CO-OPS
- NOAA OR&R

Project Description

In the last decade, IOOS has made remarkable progress toward an interoperable framework for working with ocean forecast model data. Providers deliver data via community standard OPeNDAP services using Climate and Forecast (CF) metadata conventions, and developers then use standard libraries to build applications that will work with any structured grid model (e.g. ROMS, NCOM, HYCOM, Delft3D) without the need to write any model-specific code. An outstanding gap, however, is the ability to fully support unstructured grid (e.g. triangular mesh) models, which are now being used throughout IOOS for both inundation and full 3D operational ocean forecasts.

The need for better tools to work with unstructured grids has been apparent for some time: In 2006, NOAA and NSF co-sponsored a workshop, "Community Standards for Unstructured Grids," which was attended by 30 representatives from academic, government, and commercial organizations. The consensus statement of the workshop was "The unstructured ocean model community should define a data model (in the language of OPeNDAP) and ensure the mapping of that model onto netCDF3 and HDF5 is functional". The community subsequently developed the UGRID Conventions (http://ugrid-conventions.github.io/ugrid-conventions/), which extend the CF metadata conventions to describe unstructured grid model output. It remains to embed these standards into the services most commonly used by the IOOS community.

While IOOS-supported services can be delivered and consumed by a variety of software components, Unidata's THREDDS Data Server (TDS) is the most widely used in IOOS. The TDS not only provides OPeNDAP access, but provides the OGC Web Map Service to create maps and NetCDF Subset Service to extract datasets based on longitude/latitude and calendar date extents. The Unidata NetCDF-Java library, which underlies the TDS, also supports many client applications (e.g. Unidata IDV, Matlab NCTOOLBOX, University of Reading NcWMS, NASA Panoply).

IOOS COMT 1.0 supported the development of the UGRID Conventions and the development of pilot UGRID-compliant tools, such as the pyugrid python library and the prototype ugrid branch of the NCTOOLBOX Matlab toolbox. The NCTOOLBOX work showed the promise of supporting UGRID in NetCDF-Java, enabling interoperable access to water level predictions -- a strong focus in COMT 1.0. The code was not integrated into the Unidata code base, however, and significant technical and code development (with documentation, testing, support) are still needed to bring unstructured grid models to the same level of functionality as structured grid models.

We propose to build on the success of this framework and these proof-of-concept demonstrations, extending the NetCDF-Java library to allow unstructured grid models to be "first-class citizens" in the Unidata software stack. While Unidata developers have long been interested in developing these capabilities, Unidata is primarily driven by the needs of atmospheric forecast model users. Unidata's technical expertise as developers of the THREDDS technologies, coupled with Axiom Data Science's working relationship with the IOOS Regional Association partners, will help ensure that this gap in functionality is closed

and that the new features are communicated to potential users efficiently. We hope COMT will help bring this long-needed functionality to the IOOS community.

Background

Project Participants

This project will rely on the contributions of three main participant groups:

- The Unidata program, which developed and maintains the core software technologies slated for enhancement, and which will be responsible for the bulk of the software development work. For more than 30 years the Unidata program — one of the community programs of the University Corporation for Atmospheric Research (UCAR) — has worked in concert with the geoscience education and research community to develop and provide innovative data systems, tools, techniques, and resources to support data-enabled science to understand the Earth system. Both Unidata and UCAR are primarily sponsored by the National Science Foundation.

- Axiom Data Science, which works closely with IOOS Regional Association partners, and which will be responsible for assisting users with deployment and implementation of the new technologies. Axiom Data Science is an informatics and software development firm focused on developing scalable solutions that can be leveraged across a variety of users, clients and institutions in order to make the best tools for data management, integration, and visualization.

- A core group of users of unstructured model grid products, who will serve on an advisory committee that advises on feature design, reviews project progress, and beta tests new software. (See letters of support.)

Core Technologies

The core technologies to be enhanced by this proposed work are part of the THREDDS project. THREDDS (Thematic Realtime Environmental Distributed Data Services) is a set of data technologies developed and supported by Unidata.

The THREDDS project encompasses both NetCDF-Java and the THREDDS Data Server (TDS). NetCDF-Java is a client-side java library; the TDS is a web service technology, which is built on top of NetCDF-Java. The following sections discuss both components of THREDDS, and highlights areas in which extensions are needed in order to fully support UGRID datasets.

NetCDF-Java Enhancements

The NetCDF-Java library implements a Common Data Model (CDM), which is a generalization of the NetCDF, OpenDAP, and HDF5 data models. The NetCDF-Java library is a 100% Java framework for reading netCDF and other file formats into the CDM, as well as for writing to the netCDF-3 file format. (Writing to the netCDF-4 file format requires installing the netCDF C library.) The NetCDF-Java library also implements the NetCDF Markup Language (NcML), which allows a user to add metadata to CDM datasets and to

create virtual datasets through aggregation. The THREDDS Data Server (TDS) is built on top of the NetCDF-Java library. NetCDF-Java is developed and supported by Unidata, a division of the University Corporation for Atmospheric Research (UCAR), and is sponsored by the National Science Foundation.

The NetCDF-Java library has three main architectural layers (Figure 1):

- The **Data Access Layer**, represented by the NetcdfFile API
- The **Coordinate System Layer**, represented by the NetcdfDataset API
- The **Feature Type Layer**, represented by the various scientific feature type APIs (e.g. GRID, IMAGE, POINT)

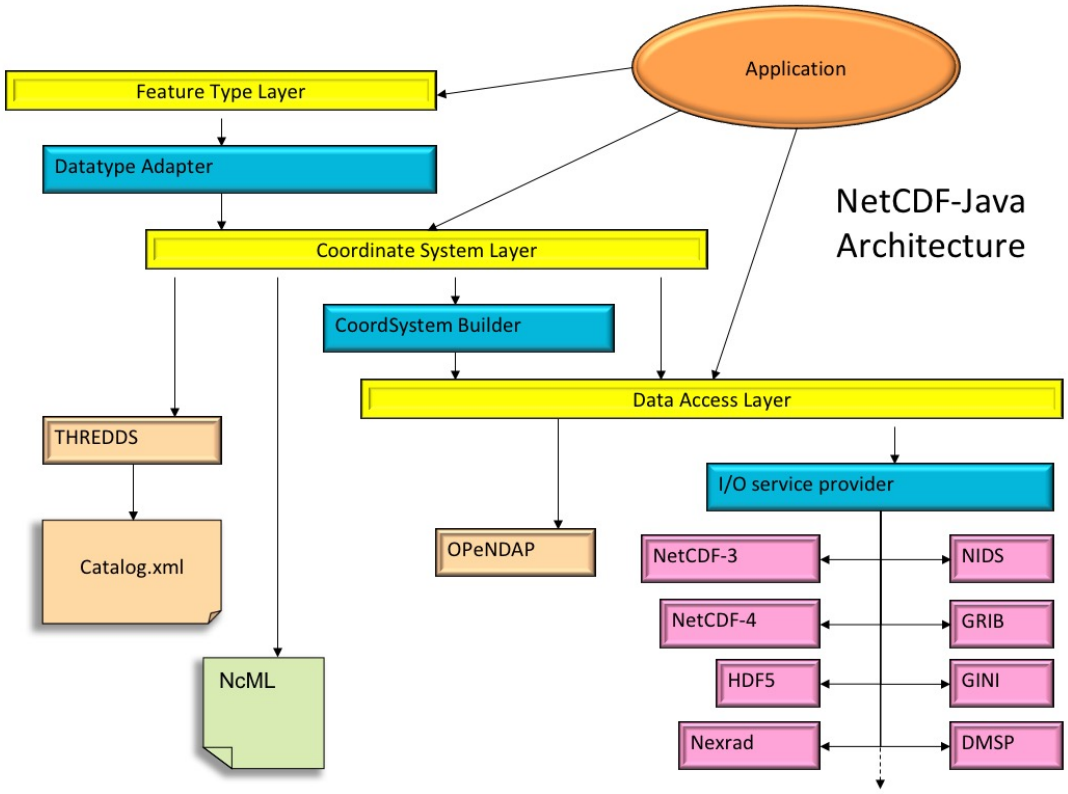

Figure 1: NetCDF-Java architecture. The three layers of dataset support are highlighted as yellow boxes. UGRID extensions are needed at the Coordinate System and Feature Type layers.

Data Access Layer

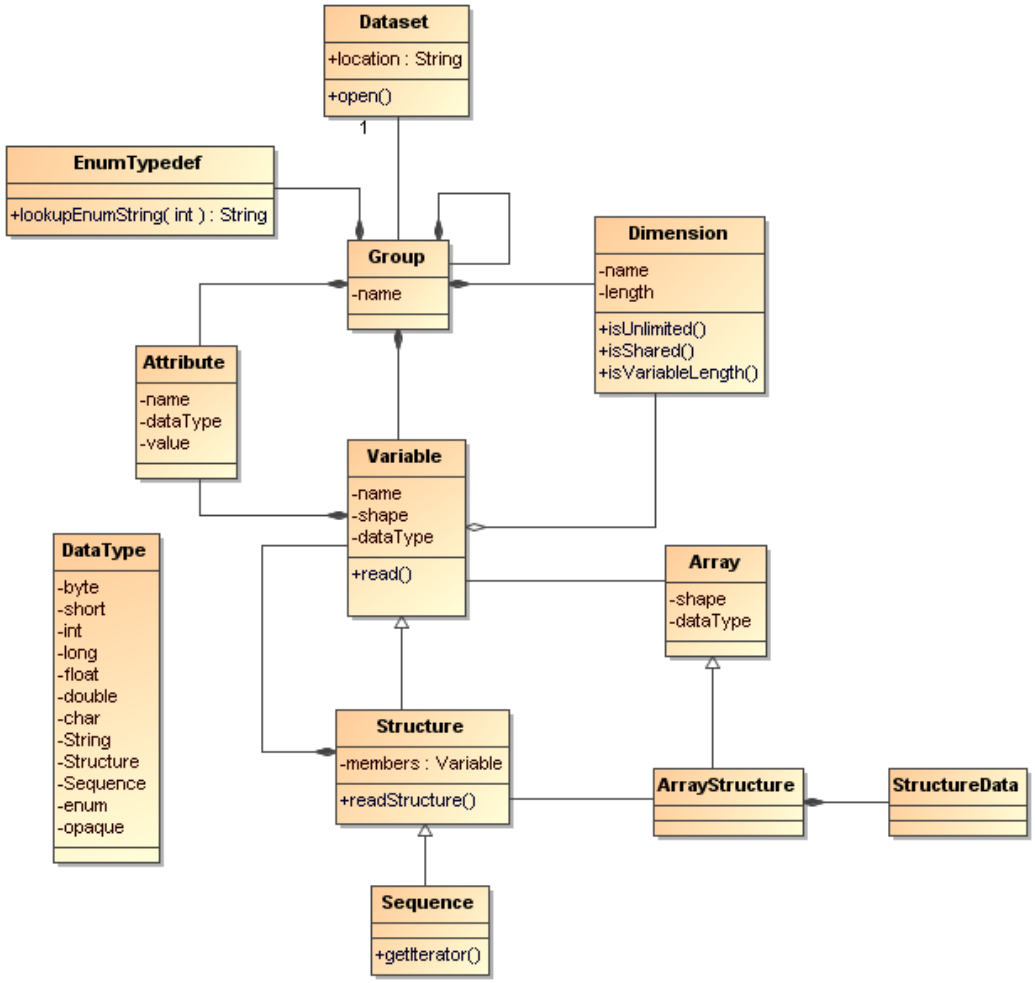

Figure 2: The Common Data Model (CDM), which is a generalization of the NetCDF, OpenDAP, and HDF5 data models.

The Data Access Layer provides is most basic level of NetCDF-Java with a consistent access end-user API for library users. It presents the data contained in a file on disk through the Common Data Model (Figure 2).

Proposed Extensions to the Data Access Layer

UGRID is currently supported at the Data Access Layer, in that the most basic level remote access to the underlying file is enabled (array index based access on a per-variable basis).

Coordinate System Layer

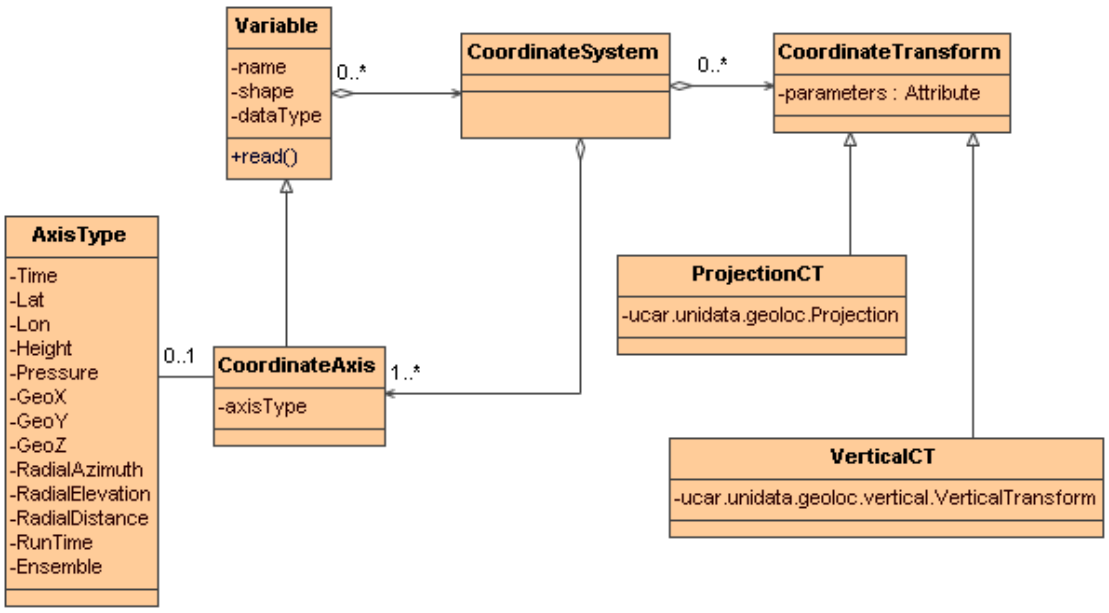

Figure 3: Coordinate Systems, as used in the Coordinate System Layer of NetCDF-Java.

The Coordinate System Layer is the next abstraction layer in NetCDF-Java (Figure 3). At this layer, variables from the Data Access Layer can be enhanced to represent Coordinate Axes and, potentially, Coordinate Transforms. Coordinate Axes are Variables that can be classified as variables containing the real date/time of the dataset, latitude and longitude coordinates, height (or pressure) vertical coordinate. When paired with a Coordinate Transform, the raw grid x and y values can be transformed into latitude and longitude values. Vertical transformations can be applied as well. Other Coordinate Axes are recognized, such as Radial Azimuth, Radial Elevation and Radial Distance for polar coordinates, and RunTime and Ensemble for forecast model output data.

Proposed Extensions to the Coordinate System Layer

Work must be done to support the coordinate systems contained within unstructured grids, which will bring UGRID up to the Coordinate System layer support. This will enable users of the NetCDF-Java library to utilize UGRID datasets with a higher-level of understanding, as each data variable will have attached coordinate system information as part of the NetcdfDataset java object. This will also provide the final stepping stone to reach the highest level API of NetCDF-Java, which will result in the library fully supporting UGRID datasets.

Feature Type Layer

The Feature Type layer is the highest-level API in NetCDF-Java. At this level, data are categorized as Feature Type objects, which represent recognizable data configurations such as a grid, a swath, a point, or a trajectory. Using the Feature Type APIs, users can extract

subsets of the Feature Type objects in geophysical coordinate space -- that is, using time strings, latitude, longitude, and or height/depth. (By contrast, the Data Access layer APIs allow subsetting only in array index space, which requires a client application to know how to map array indices into coordinate values.)

The NetCDF-Java library supports the following Feature Types: GRID, RADIAL, SWATH, IMAGE, POINT, PROFILE, SECTION, STATION, STATION_PROFILE, and TRAJECTORY.

Proposed Extensions to the Feature Type Layer

Providing UGRID functionality will require either an extension of the existing GRID Feature Type or the creation of a new UGRID Feature Type. This addition will allow users of NetCDF-Java to subset UGRID datasets using temporal and geophysical coordinates and to use all other TDS services in the same way they can with currently-supported Feature Types.

THREDDS Data Server Enhancements

The THREDDS Data Server (TDS) is a web server that provides metadata and data access for scientific datasets, using OPeNDAP, OGC WMS and WCS, HTTP, and other remote data access protocols. TDS also includes a service (the NetcdfSubset Service) for subsetting based on temporal and geophysical (latitude / longitude / height) queries.

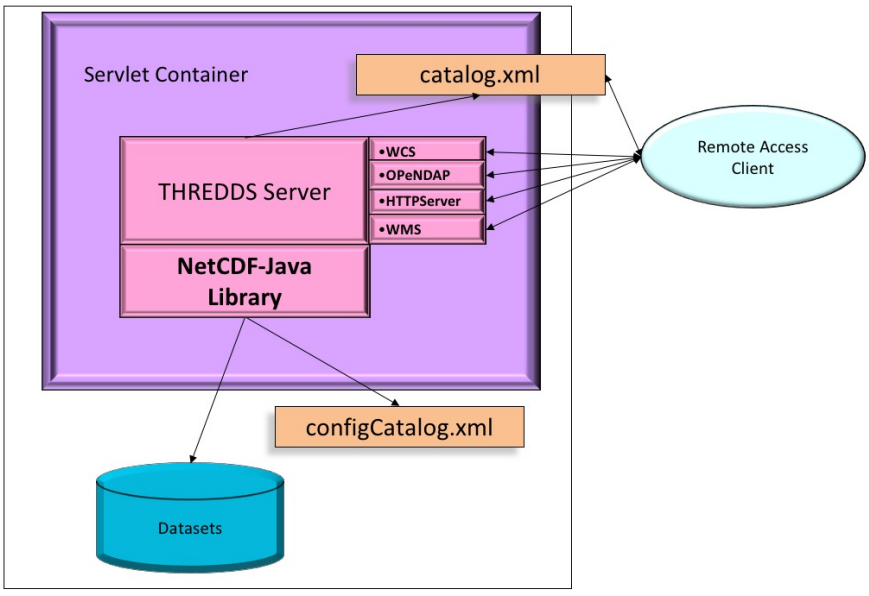

Figure 4: Diagram of some of the components of the TDS. Note that the core of the TDS is the NetCDF-Java Library.

Some of the technological components of the TDS:

- THREDDS Dataset Inventory Catalogs are used to provide virtual directories of available data and their associated metadata. These catalogs can be generated dynamically or statically.
- The NetCDF Markup Language (NcML) allows users to modify and create virtual aggregations of CDM datasets.
- An integrated server provides OPeNDAP access to any CDM dataset. OPeNDAP is a widely used subsetting data access method that extends the HTTP protocol.
- An integrated server provides bulk file access through the HTTP protocol.
- An integrated server provides data access through the Open Geospatial Consortium (OGC) Web Coverage Service (WCS) protocol, for any "gridded" dataset whose coordinate system information is complete.
- An integrated server provides data access through the OGC Web Map Service (WMS) protocol, for any "gridded" dataset whose coordinate system information is complete. THREDDS integrates versions of two technologies developed at the University of Reading: NcWMS and its companion "Godiva" web application for WMS access and online visualization.
- The integrated ncISO server provides automated metadata analysis and ISO metadata generation.
- The integrated NetCDF Subset Service allows subsetting certain CDM datasets in coordinate space, using a REST API. Gridded data subsets can be returned in CF-compliant netCDF-3 or netCDF-4 format. Point data subsets can be returned in CSV, GeoCSV, XML, or CF-DSG netCDF files. WaterML is supported for some datasets.

The TDS is implemented in 100% Java, and is contained in a single war file, which allows very easy installation into a servlet container such as the open-source Tomcat web server. Configuration has been made as simple, automatic, and secure as possible. As with NetCDF-Java, writing netCDF-4 files requires the use of the netCDF-C library.

The TDS is built on top of NetCDF-Java, and as such, can remotely serve any dataset that can be read by the library. However, the types of services through which the underlying data can be exposed depends on the layer of NetCDF-Java that is supported. For example, CF-compliant netCDF files can be read into the Feature Type layer as GRIDs, and therefore can be served out using any service of the TDS that supports GRID data types (OPeNDAP, WMS, WCS, ncISO, NetCDF Subset Service, etc.). Currently, UGRID data are only supported at the Data Access Layer, which means very few TDS services can expose the data (e.g. OPeNDAP, ncISO).

The TDS is capable of creating virtual aggregations of netCDF files in two ways: NetCDF Markup Language (NcML) aggregations, and Forecast Model Run Collections (FMRCs). NcML aggregations require intimate knowledge of the underlying data. Coordinate variables and data variables are joined explicitly by the user, and these variables can be different between different models or even different versions of the same model. NcML aggregations can be time consuming to construct, and the performance is acceptable only for collections on the order of tens of files because each aggregation is constructed at the time data is requested. While NcML aggregations have a caching capability, the cache is very unreliable

in its current iteration, and updating the cache can be time consuming. As a result of all of these factors, NcML aggregations are fragile and difficult to maintain, especially when used with inherently complex UGRID coordinate systems. FMRC aggregations are created programmatically and require less user input, which reduces the fragility of the aggregation. However, performance of FMRC aggregations suffers significantly for collections on the order of 100 files.

A newer, third way of aggregating data is the Feature Collection. Much like FMRCs, Feature Collection aggregations are done programmatically with little user input. Coordinate systems found within datasets are aggregated automatically based on rules defined by standards documents (i.e. CF, UGRID). Currently, the TDS exposes both FMRCs and collections of GRIB messages via the Feature Collection mechanism. While FMRCs have not been optimized to scale to large collections, GRIB Feature Collections have, and provide a pattern for how to enable new, scalable Feature Collections for both GRID and UGRID collections.

Proposed Extensions to the TDS

GRID Feature Collections will be patterned off portions of the scalability work done for GRIB Feature Collections. Some NetCDF-Java objects are serialized to disk as .ncx files (referred to as CDM Index files) using Google's *protocol buffers* library. Each data file has a CDM Index File that contains the coordinate system information associated with the data file. The CDM index files are several orders of magnitude smaller than their corresponding data files, and result in very little I/O overhead when accessed by the TDS. In turn, collection level object is constructed and stored in an .ncx file, which is called the GRID Collection Index file. The GRID Collection File can be an order of magnitude or more smaller than the CDM Index Files. When the TDS receives a data request, the collection level index will be opened and unserialized. The TDS is then able to quickly determine which data files contain the data needed to satisfy the request. Next, the index files belonging to the data files which satisfy the request are opened by the TDS; these files allow the TDS to quickly extract the requested data from the data files on disk and fulfill the request. When a new data file is added to the collection, a new CDM Index File corresponding to the new datafile is created, and the GRID Collection Index file is updated. The process of creating a new CDM Index File and updating the Collection Index File is very efficient, and allows for new data to be exposed to end users quickly.

The creation and management of the index files for GRIB Feature Collections is handled by a separate process called the THREDDS Data Manager (TDM); index files required for GRID Feature Collections will be managed by the TDM as well.

Summary of Technical Enhancements

In order to support model output following the UGRID specification at the level at which CF-compliant model output are supported, extensions to both NetCDF-Java and the TDS need to be made. Specifically, the extensions necessary are listed under the "Proposed Extensions" headings under "NetCDF-Java Enhancements" in this proposal, and are set to be addressed in year one of the proposed work.

However, while this will indeed allow support for UGRID, the scalability of aggregations of large numbers of model output files will still be lacking and present performance issues, as is presently the case with collections of CF-compliant model output. The extensions necessary to address the scalability issues are listed under "Proposed Extensions" headings under "THREDDS Data Server Enhancements" in this proposal, and are dependent on the previous work to enhance the NetCDF-Java libraries. The scalability-focused work of this proposal is set to be addressed in year two.

Audience

Any provider or user of unstructured grid model data will benefit from the work done as a result of this proposal, but its specific focus is on the IOOS Federal Backbone operational partners and Regional Association partners. These groups have been struggling to find effective ways to serve their unstructured grid model output.

IOOS partner NOAA CSDL uses the unstructured ESTOFS model (ADCIRC) for storm surge prediction on both the Atlantic and Pacific CONUS. IOOS Partner NOAA CO-OPS uses the unstructured FVCOM and SELFE models for coastal forecasts in Lake Erie, the Gulf of Mexico, San Francisco Bay and the Columbia River. IOOS Regional Associations SECOORA and NERACOOS also use FVCOM models to provide regional forecasts for their stakeholders (GLOS transitioned their FVCOM model of Lake Erie to CO-OPS).

Approach

Objective 1 (Year 1)

Supporting unstructured grids (UGRID) models in all three layers of NetCDF-Java (Data Access Layer, Coordinate System Layer, and Feature Type Layer).

Add Coordinate System layer support for unstructured grids:

- Identify collection of representative files (from various models). Input from partners / IOOS partners needed to identify and obtain sample files.
- Create unit and integration tests based on expected geophysical coordinate properties of the unstructured grids represented by the sample collection. Expected properties derived from input of partners.

Investigate addition of UGRID to GRID Feature Type:

- Identify areas in which GRID Feature Types fall short of supporting unstructured grids.
- Expand GRID Feature Types, or create a new UGRID Feature Type, based on GRID Feature Types. Anticipated that this will require the bulk of time needed to complete objective 1.
- Create unit and integration tests to develop and exercise subsetting capabilities for UGRID files.
- Create basic maps of various fields within UGRID files to visually verify geolocation of unstructured grid fields. We will work closely with our IOOS partners in this verification process.

Upon the completion of Objective 1 at the end of quarter 3 of year 1, a version of TDS, complete with the enhancements above, will be delivered to IOOS Regional Association partners by Axiom Data Science. Configuration catalogs will be created by Axiom Data Science will use the new UGRID FMRC capabilities to serve out unstructured grid model output.

Objective 2 (Year 2)

Improve netCDF aggregations in the THREDDS Data Server. (Dependent on successful completion of Objective 1).

- Abstract out GRIB specific portions of the Collection Level Index files used by the TDS, leaving behind a generic GRID Collection Level Index interface, extended by GRIB.
- Extend the GRID Collection level Index interface, if needed, to support UGRID.
- Extend the types of collections exposed by the new GRID Feature Collection to match those exposed by FMRC (i.e. Add Constant Forecast Offset, Constant Forecast Date).
- Verify the new GRID Feature Collection capabilities as exposed by the TDS. We will work closely with our IOOS partners in this verification process.

Upon the completion of Objective 2 at the end of quarter 3 of year 2, a version of TDS, complete with the enhancements above, as well as the THREDDS Data Manager (TDM), will be delivered to IOOS Regional Association partners by Axiom Data Science. In coordination with Unidata, configuration catalogs created at the end of year 1 will be updated by Axiom Data Science to use the new GRID Feature Collection capabilities to serve out unstructured grid model output. FMRC collections of regular gridded output will be updated to use the new GRID Feature Collection as well.

Throughout the project, we will hold quarterly virtual meetings with an Advisory Committee of our IOOS partners. The meetings will provide project updates, as well as solicit feedback and guidance on software development and new features. IOOS partner participation will be critical in identifying and compiling a collection of representative UGRID output files, providing guidance on how unstructured grids map into geophysical space, identifying the expected geophysical grids for the representative UGRID files, and helping exercise and validate the new UGRID subsetting capabilities from an end-user point of view. Members of the Advisory Committee will also work with Axiom Data Science to install and test the new services in their own computing environments, providing direct feedback from a data provider's point of view. As shown in the letter of support, our partners have agreed to provide the resources to accomplish these critical tasks by assigning a designee to both serve on the Advisory Committee that reviews progress quarterly and to beta test new UGRID features of the TDS.

Benefits

The software enhancements to the THREDDS Data Server and NetCDF-Java libraries will benefit all IOOS Federal Backbone partners and Regional Associations who provide unstructured grid forecast model forecasts, and will benefit all users seeking to access these forecasts.

Once UGRID support is incorporated into the TDS, data providers will be able to simply place their forecasts into a directory when they become available; the TDS Forecast Model Run Collection (or GRID Feature Collection, if funded) will automatically produce virtual datasets that contain the Best Time Series dataset that most users want, as well as a double time coordinate (time and forecast hour) dataset that is essential for determining how forecast skill changes with increasing forecast time.

Attached to this proposal is a letter of support signed by both operational and regional association partners indicating that they will directly benefit from this work.

Milestone Schedule

Table 1 on the following pages shows the project milestone schedule.

Note on Technical Readiness Levels

We consider NetCDF-Java and the THREDDS Data Server in its current implementation to be RL 9, as they are currently in operational use by numerous organizations, including IOOS Regional Associations.

Current UGRID support in the NetCDF-Java library and in the TDS is RL 4 (intermediate phase of development); by the end of the project, UGRID support will incorporated into both the NetCDF-Java library and the TDS at RL 8 (finalized service or tool).

Table 1: Project Tasks / Milestones

Objective	Activity	Task / Milestone	Year 1 Q1	Year 1 Q2	Year 1 Q3	Year 1 Q4	Year 2 Q1	Year 2 Q2	Year 2 Q3	Year 2 Q4	Deliverable
Supporting unstructured grids (UGRID) models in all three layers of NetCDF-Java (Data Access Layer, Coordinate System Layer, and Feature Type Layer).	Add Coordinate System layer support for unstructured grids	Identify collection of representative files (from various models)									
		Create unit and integration tests based on expected geophysical coordinates of the unstructured grids represented by the sample collection.									**Code:** Unit tests posted to THREDDS github repository and exercised by Unidata Continuous Integration pipeline.
	Investigate addition of UGRID to GRID Feature Type	Identify areas in which GRID Feature Types fall short of supporting unstructured grids									**Report:** UGRID as a GRID Feature Type in Netcdf-Java.
		Expand GRID Feature Types, or create a new UGRID Feature Type, based on GRID Feature Types.									**Documentation:** Document extensions to GRID (or new UGRID FeatureType).
		Create unit and integration tests to develop and exercise subsetting capabilities for UGRID files.									**Code:** Unit tests posted to THREDDS github repository and exercised by Unidata Continuous Integration pipeline.
		Test UGRID FMRC support on Unidata TDS. Netcdf-Java support at RL 7.									Publicly accessible UGRID test datasets on operation Unidata server.
		Create basic maps of various fields within UGRID files to visually verify geolocation of unstructured grid fields									**Report:** Basic UGRID visualization tests.

Table 1: Project Tasks / Milestones					
Objective	**Activity**	**Task / Milestone**	**Year 1**	**Year 2**	**Deliverable**
	Deployment to IOOS Regional Associations	Deploy new TDS utilizing NetCDF-Java enhancements to use FMRC aggregations on UGRID collections. Netcdf-Java support at RL 8.			**Product**: TDS war for deployment.
Improve netCDF aggregations in the THREDDS Data Server.	TDS Aggregation Enhancements	Abstract out GRIB specific portions of the Collection Level Index files used by the TDS, leaving behind a generic GRID Collection Level Index interface, extended by GRIB			**Code:** Verify new code and abstractions using Unidata Continuous Integration test pipeline.
		Extend and test the GRID Collection level Index interface, if needed, to support UGRID			**Code:** New code and unit/integration tests posted to THREDDS github repository and exercised by Unidata Continuous Integration pipeline.
		Extend the types of collections exposed by the new GRID Feature Collection to match those exposed by FMRC (i.e. Add Constant Forecast Offset, Constant Forecast Date). TDS UGRID Support at RL 8.			**Code:** New code and unit/integration tests posted to THREDDS github repository and exercised by Unidata Continuous Integration pipeline. **Documentation:** Document extensions to GRID (or new UGRID FeatureType).
		Test UGRID GRID Feature Collection support on Unidata TDS. TDS UGRID Aggregation support at RL 7.			Publicly accessible UGRID GRID Feature Collection test datasets on operation Unidata server.

Table 1: Project Tasks / Milestones

Objective	Activity	Task / Milestone	Year 1	Year 2	Deliverable
	Deployment to IOOS Regional Associations	Deploy new TDS utilizing NetCDF-Java enhancements to use GRID Feature Collection aggregations on UGRID collections. TDS UGRID Aggregation support at RL 8.			**Product**: TDS war for deployment.
Meetings / Stakeholder Outreach *	Community Meetings	COMT Annual Meeting - year 1 (actual date TBD)			**Report:** Materials assembled for annual COMT Meeting.
		2019 AGU			**Presentation:** Communicate project results to date to the community.
		COMT Annual Meeting - year 2 (actual date TBD)			**Report:** Materials assembled for annual COMT Meeting.
		Project Advisory Committee Meetings **			**Report:** Committee meeting notes, hosted in publicly accessible format.
	Project Team Meetings	Project Kickoff Meeting in Boulder, CO			**Report:** Meeting notes.
		Year 2 Kickoff Meeting in Boulder, CO			**Report:** Meeting notes.
	COMT Engagement ***				**Report:** Reports to COMT, as required.

* Also includes deliverables titled **Report:**

** Project Advisory Committee made up of individuals from organizations listed in the letter of support.

*** Also includes COMT Annual Meetings

Project Budget

Table 2 summarizes the project budget. Please see the attached Budget and Budget Narrative for complete details.

TABLE 2 REMOVED FROM THIS VERSION OF THE DOCUMENT.

Data Management Plan

This proposal seeks funding to create a unified framework for IOOS model data, extending Unidata's NetCDF-Java library and THREDDS Data Server (TDS) to fully support the unstructured grid models now being widely used in the IOOS community. This will enhance the existing COMT cyberinfrastructure framework by fulfilling a need that has existed for more than a decade, and will produce services and tools that have direct, immediate benefit to the operational IOOS Federal Backbone community as well as IOOS Regional Association partners.

Environmental Data Collected or Generated by the Project

This is a software project, so no environmental data or information will be collected or created as a part of this project.

Software Created by the Project

The NetCDF-Java and TDS software that this project enhances are Open Source (MIT-like license) and freely available to the public from the Unidata Program Center. Those interested in accessing the software or source code, or in joining the larger THREDDS development effort as an open source contributor, can find information on Unidata's THREDDS Data Server at: https://doi.org/10.5065/D6N014KG

All development will be conducted in an open manner using the THREDDS project github repository (https://github.com/Unidata/thredds) for both source code management and issue tracking, as well at the https://gitter.im/ service, which will enable real-time chat and messaging for the project team and broader open source community.

Coordination with IOOS DMAC Group

The IOOS DMAC is very familiar with the THREDDS Data Server, NetCDF-Java, and the need to support UGRID Conventions. Project subcontractor Kyle Wilcox is a DMAC Representative and can make sure that coordination with DMAC is maintained.

Data Management Point of Contact

The project PI (Sean Arms, UCAR/Unidata) will be the project's point of contact for data management and software access issues.