# Distributed Oceanographic Match-up Service (DOMS) User Interface Design

Shawn R. Smith[1], Jocelyn Elya[1], Adam Stallard[1], Thomas Huang[2], Vardis Tsontos[2], Benjamin Holt[2], Steven Worley[3], Zaihua Ji[3], and Mark A. Bourassa[1,4]

[1]Center for Ocean-Atmospheric Prediction Studies, Florida State University, Tallahassee, FL
[2]Jet Propulsion Laboratory/California Institute of Technology, Pasadena, CA
[3]National Center for Atmospheric Research, Boulder, CO
[4]Earth, Ocean, and Atmospheric Science Department, Florida State University, Tallahassee, FL
Contact: smith@coaps.fsu.edumailto:smith@coaps.fsu.edu

## Introduction

The DOMS prototype envisions two types of user access: a web-based graphical user interface (GUI) to support man-to-machine queries and a web-service application programming interface (API) to support machine-to-machine queries. The former will generally support one-off data match-ups and will allow a user to refine his search criteria interactively prior to making a request to receive the matched dataset. This approach will minimize data traffic over the network between the distributed hosts, only moving the matched data and associated metadata once the user is content with his search results. The second set of web service APIs will support routine and repeated data matching request (e.g., someone wishing to match the latest satellite data to new in situ data on a monthly basis). In addition to providing web services, DOMS will provide web tools that will aid the user in developing his web service queries. This will make it easier for a user to develop the proper syntax for his data request. Finally, we envision users combining the functionality of the GUI and API, using the former to refine their search criteria (say for a given region, month, and datasets) and then using the latter to run similar queries in a batch process mode (for example, searching the same region and datasets, but for different months).

Additional information for how to use the GUI and web services will be provided in a separate user document. This will be developed along with any help menus during the GUI and web service implementation.

## DOMS Terminology

During the initial design of the UI and API, it became necessary to establish a common vocabulary. The following terminology will be used in this document and within the UI and API.

- Primary Dataset - the dataset that must appear in each matched pair and around which the match-up tolerance is centered when looking for matches.
- Match-Up Datasets - datasets which the user wants to match to the primary dataset. May have more than one match-up dataset.
- Search Domain -The (3D) spatial region and temporal range in which the user wants to find matches. This is used to subset the primary and match-up datasets.
- Match-Up Tolerances - The radius (in horizontal space), +/- depth, and time range which defines how close a match-up point must be to a primary point in order for those two points to be considered a “match.” Used to define matches, not for creating subsets of the data within the search domain. Data are accepted to a “match” when their location, depth, and time are less than or equal to the tolerance values.
- Match-Up Parameter – The variable chosen by the user that will be compared between primary and match-up datasets. For the prototype the options are sea temperature, salinity, and winds.

- Ancillary Parameter – Variables that are not part of a chosen variable to match, in some cases these parameters may not even be indexed by the DOMS service. For example, many in-situ records include pressure, humidity, or radiation values, while satellite datasets contain information on radiances, backscatter, orbit numbers, scan lines, etc.

**Web GUI**

*User query options*

The user will be able to make a query to DOMS by selecting the match-up parameter, primary and match-up datasets, a search domain that includes a date range and geospatial bounds, match-up tolerances (horizontal space, depth, and time), platform, and data quality. Design criteria for each of these query options is outlined below.

Parameters: In the DOMS prototype, the user will have the option of selecting sea surface temperature (SST), sea surface salinity (SSS), or winds as their data types to create matches. Users will only be able to create matches based on their selected parameter in the prototype (e.g., SST primary dataset values to SST match-up dataset values). Users may match on one of the parameters and request the other two parameters if they're available. For example, a user requests wind data to be matched between a primary satellite dataset and three match-up in situ datasets and also wishes to receive SST data for each match. In this case wind data MUST exist for each satellite to in situ data pair, but SST is not required for each data pair (but will be provided if it exists for either the satellite or in situ datasets). The prototype UI will provide an option for the user to request ancillary parameters in their output files, but will not support provision of parameters that are not indexed to support data matching (e.g., humidity, air temperature, pressure, radiances, etc.).

Data Source: The user will be able to select from a list of satellite and in situ datasets that include the parameters (see above) from various sources (e.g., MURS, ASCAT, ICOADS, SAMOS, SPURS, etc.). The user will select a primary data source to which one or more match-up datasets will be paired. Once the primary data source is selected, the user will be provided a list of available match-up sources and will then select from this list. On the front end (GUI), this may be accomplished by providing a two-column structure where the primary to match-up source matches are assign across the two columns. For prototype, the UI will enforce selection of a singular (primary) data source to force 1-to-1 or 1-to-many pairing of the chosen parameter to the selected match-up datasets. In the backend (especially the data matching algorithm), the suggested column order should not have any effect on the resulting pairwise dataset (in fact using the smaller in-situ set as the primary dataset in the data matching algorithm is likely most efficient). To further refine data source selection, the GUI should provide date ranges for each source dataset available within DOMS.

Temporal Search Domain: The user will have the option to enter a start date and end date to define the range of days for which they are seeking matched observations. For the prototype, a single day will be the finest temporal granularity that a user can select (i.e., sub-daily time ranges will not necessarily be supported). The GUI should employ both a graphical calendar and text entry windows for providing start and end dates. To aid user selection of date range, the GUI should only allow start/end date entry for the valid range of dates for the data sources

and parameter previously selected. Other options to be considered during prototype development include determining whether or not (1) a maximum date range will needed (based on match-up algorithm speed) and (2) the data density within the user selected spatial domain (see below) could limit the range of dates in a single match-up query.

Geospatial Search Domain: The user will define a three dimensional volume in which they are seeking matched data. The horizontal dimensions will be defined using a range of latitude and longitudes and in the prototype will be selected using a "string" box or text entry of the maximum and minimum latitude and longitude values. Text entry will use a -180˚ to 180˚ convention for longitude and a -90˚ to 90˚ convention for latitude. Ideally the UI would draw the string box as the user types in the text entry boxes. The ability to support polygons to define the horizontal domain (e.g., supporting geolayers so that a user can select an ocean basin or ocean feature; http://www.marineregions.org/) was considered, but will not be implemented in the prototype.

The user will also be able to select a depth range in whole meters using, at a minimum, a text entry box. For the prototype, the depth will be limited to 20 m below the ocean up to 200 m above the surface (for wind measurements from large merchant vessels). Users will be required to enter a depth range (no default value). The UI should automatically swap between requesting a "depth" range for ocean parameters (SST, SSS) and a "height" range for atmospheric (wind) parameters. These values will be converted on the backend to depth values to support queries to the index and for data matching. In addition, when a measurement has no depth/height in the index, it will be "discovered" in the data query to the index and in the matching algorithm. A clear message should be provided on the UI noting that "values with missing depth will be matched to all specified depth ranges".

Data Match-up Tolerances: Users will be required (no default value) to input a spatial radius in meters, a depth range +/- the primary dataset observation depth, and a temporal range +/- the primary dataset observation time in minutes. Minimum requirements for entering match-up tolerances will be via a text box. Setting some minimum and maximum tolerances may be necessary as the prototype is benchmarked. For the prototype, the spatial radius will be limited to 0 to 30000 meters.

Platform: The user will be given the option to filter the primary and/or match-up datasets by platform type (e.g., vessel, buoy, drifter, glider). Whenever possible, the GUI should dynamically generate a list of platforms based on the source(s) selected by the user. When a list is presented, the option should exist (via a button or the like) to select all / none of the items on the list. Optional for the prototype would be to create a faceted list of platforms, whereby platform types could be grouped (e.g., buoy and drifter placed together). To facilitate the GUI, all platforms must be indexed using standard terms as outlined in the DOMS Software Engineering Specification and translation documents for individual datasets.

Data Quality: The user will be provided options to filter the primary and/or match-up datasets based on the quality of the observations. In the prototype, filtering by data quality will only be supported for in situ datasets. When quality indicators are available, the default data returned

will be those filtered to be “good” quality as determined (and documented in translation specifications) by the data hosts for individual datasets (e.g., ICOADS, SAMOS, SPURS). Alternatively, the user will have the option to receive all data values regardless of their quality (note: this will be the default option for satellite data with complex flag schemes in the prototype). In either case, the user will be provided with, at a minimum, the simplified data quality flags as defined by the in situ data hosts or an indicator that no data quality has been applied to the observations (e.g., for the satellite data) as part of the matched data output files. For the prototype, inclusion of the original source data quality flags in the output files is optional.

*Work Flow*

The UI should support a clear work flow that leads the user through a series of steps to retrieve matched data that meets their query options.

The work flow should begin with the selection of the search parameter and a search domain. This would be followd by selecting the primary and match-up datasets (data sources). Each higher-level selection should filter/constrain the available options for the subsequent lower-level options in the workflow. For example, the selection of a parameter should limit the options for the search domain and data sources based on the contents of the data sources available to DOMS (e.g., if you select SST as your parameter, only primary and match-up datasets containing SST should be available for selection by the user). Equally when a spatial/temporal search domain is specified, this should limit the choices for primary and match-up parameters to only those with data in the user-specified domain.

The action from the UI to search the appropriate satellite and in situ datasets for subsets of data that meet the parameter and domain criteria should not be initiated until all these selections are complete (possibly with a button to trigger that action). This allows the user to make minor modifications to their criteria without DOMS running unnecessary queries to the data hosts in the backend. Once initiated and complete, the UI should display basic counts and maps of the data available for consideration by the data matching algorithm for each match-up dataset selected.

Once the parameter, domain, and primary and match-up datasets have been selected and the initial search query run, the UI should have the option for the user to download the discovered data subsets. Although this has been considered, the subset feature will likely not be in the prototype.

Selection of the match-up tolerances should be separated from the selection of the parameter, datasets, and search domain on the UI to avoid confusion for the users. Also the match-up tolerances play no role in the first stage of the DOMS processing whereby subsets of the primary and match-up datasets are identified based on the search domain.

Once match-up tolerances are selected, the user will be given the option to run the data matching algorithm. Two data matching options will be supported: identify all matches and identify only the closest match. In addition, the user will be provided with options to filter the

matched data by platform and data quality. These filters, ideally, will be applied prior to running the data matching algorithm to reduce the volume of data being searched for primary to match-up dataset pairs.

Upon completion of the data matching operation, the user will be provided with several views of their matched data along with options to download the matched datasets. The prototype will display basic plots of the matched data, including a map of the difference in the parameter, a scatter plot of the matched parameter values, and a histogram of the differences in the parameter. When the number of matches is small, a tabular display of the primary to match-up dataset values will be provided. Output options for downloading the matched dataset will include comma separated value (CSV) and network common data form (netCDF) files. The user will be provided with links to the data files and will have options to download one or both of the output formats. Beyond the prototype, options should exist for users to submit an email address so that they can be notified when larger data matching requests are complete as opposed to waiting for the full process to complete during their active internet session.

Throughout the workflow, the user should be provided with indicators of the status of their requests. Status bars (or the like) should be provided when backend processing is underway (e.g., identifying subsets, running the matching algorithm, and creating plots/output files). Users should be clearly notified when their requests are complete or when errors occur. In addition, a “clear all” or “start over” option should exist on the UI to take the user back to the beginning of the selection criteria in case they wish to begin again. Additional back or return buttons should exist throughout the process on the UI in case users wish to modify their selections during their session on the DOMS UI.

The prototype will be designed to allow submission and completion of only one request by a user in a single internet session. Future development will consider how to manage and display multiple, parallel match-up requests.

*Output Files*
As noted above, the UI will support the creation and downloading of both CSV and netCDF files.

The CSV files will include a common section where each record will include the time, latitude, longitude, depth, parameter value, quality flag, source, and platform for the primary dataset followed by the same eight values for one of the match-up datasets. Each column in the CSV file will be labeled in a header with the units clearly noted. Following the common section, a data source specific “meta” string may be included. These meta strings will contain additional values and information that the data provide feels are useful to the interpretation of the matched dataset and how to decode each meta string will be included in the data source documentation.

The CSV file will also contain a header (similar to netCDF global attributes) that will contain a detailed summary of the query parameter used to create the matched dataset in the file. Optionally, the full query URL may be included in the header information. The header should also contain additional metrics for the query (e.g., date file created, time to run, DOMS version,

etc.) and appropriate links to the documentation for the primary and match-up datasets (via a DOMS documentation URL).

Data in each CSV file will be sorted by time, latitude, and longitude based on the primary dataset values. Data values will be output as they are stored in each host's Solr index. Wind data will include the u, v, and speed values (no direction) in the output files, unless directional information is not available in which case u and v will be missing.

For depth, some special conditions need to be applied. When depth is missing in the Solr index for a dataset, these records are still included in the data search and match-up by default; however, the depth value in the output file will be missing so that the user know no depth information existed for that primary or match-up data value. Ideally, depth and height will be output with values that are clearly understandable to the user, though in the prototype simply outputting the depth as positive downward relative to the ocean surface (so that heights above the ocean are negative) is anticipated. Documentation needs to make this clear to a user.
The netCDF files will contain the same information as the CSV files. Whenever possible, the netCDF files will use the Climate and Forecast (CF) conventions. CF names for variables, dimensions, and attributes will be outlined during the prototype development.

Future developments may consider providing an aggergated statistical representation of the matched data (as opposed to all the paired data). This may become important for large data matching operations where the individual paired data may prove too large for the user to manipulate. Examples may include averages, variances, medians, etc. of the multiple match-up values that are paired with a single primary value.

**Web Service Portal**

The web services provided via an applications programming interface (API) will support all of the search, subsetting, and data matching capabilities listed under the graphical user interface. As needed, the APIs will be developed to run different stages of the workflow outlined for the UI. For example, APIs will be provided to query the status of the DOMS service and data nodes, create data subsets, identify matched data pairs, and create data statistics and plots. Depending on the API results, output will be provided in CSV, netCDF, and or JSON. Each web service will be fully documented on the DOMS website and the documentation will include the range of valid input values for each field (using controlled vocabularies whenever possible). Sample calls for each API will be provided and a web-based tool that supports query syntax development should be developed as part of future API documentation.