Rosetta

A white paper on the challenges of sharing observational datasets

The Challenges of Sharing Observational Data

Observational studies are a critical component of geosciences education and research. Scientists and students often collect observational data in the field using dataloggers, which interface with various types of instrumentation to sample and store data at the observation site. The data are held in the datalogger until they can be collected during a site visit or transmitted over a network. The data obtained from dataloggers are almost always in an ASCII-based format, such as a comma separated value (csv) file, facilitating use by scientists.

Each type of datalogger produces a unique form of ASCII file; sometimes the differences between files produced by different dataloggers are quite significant. The result is that *N* ASCII-based data files will likely have *N* different layouts (even within the same project). This presents an immediate roadblock to outside users attempting to integrate the collected datasets into their own scientific workflows.

Data providers often transfer basic knowledge of the dataset to others by providing a README file, as the ASCII files collected from the field often lack even the most basic metadata (such as the units of the data being sampled). The format and quality of the README files varies greatly between projects (and sometimes even within the same project), and even the best README files may omit important metadata.

While a particular ASCII file layout may work well in the data providers' scientific workflow (i.e. analysis and visualization tools), other users must spend significant effort to get the data into a format that fits their workflow. Furthermore, data collected from the field often end up in a spreadsheet file, which presents another hurdle for users who do not do analysis using spreadsheet software.

The plethora of different ASCII and spreadsheet formats makes it very difficult to provide standard web services to access these datasets. For example, aggregating multiple files or spatially and/or temporally subsetting datasets quickly becomes impractical, because special code must be written allowing a data portal to handle each ASCII format or spreadsheet layout.

This whitepaper describes the approach taken by the *Rosetta* project at the Unidata Program Center to improving the quality and accessibility of observational data sets while minimizing disruption to existing scientific workflows.

Possible Solutions

The barriers discussed above can be addressed and greatly reduced by requesting that datasets meet certain quality standards if they are to be shared. (Note that sharing of data is required for NSF-funded projects). This section describes three data formats of varying quality, along with some pros and cons of using these formats to share observational data

- *User Defined ASCII format (The status quo: each project defines its own format)*
  - Pros:
    - Easy to read and write
    - Similar steps used to access data across a wide range of tools/languages
    - Works well in the data providers existing scientific workflow
  - Cons:
    - *N* sample files likely to have *N* different layouts
    - Reading these files can become time consuming
    - Basic metadata often not included in the datafile
    - Difficult (and sometimes impossible) to provide standard services (aggregation, subsetting, etc.), which allow other users to efficiently access the data.

- *Standard Layout ASCII format*
  - Pros: Same pros of the *User Defined ASCII format* from above, plus:
    - Standard file layout allowing users to know what to expect when dealing with new dataset
    - Easy to visually inspect file through the use of a text editor or spreadsheet software
    - Metadata included with file in a standard layout (self describing file)
  - Cons:
    - Relies on data providers to consistently follow a specification, leading to frustration for the data provider unless he or she is an expert in the specification.
    - Difficult, although easier than the User Defined ASCII format, to implement standard services (aggregation, subsetting, etc.)

- *A Standard, self-describing, machine-independent data format*
  - Pros:
    - Easy to provide services for end users (aggregation of files, subsetting, etc.)
    - Metadata embedded in a standard way (self describing file), which results in the need for a README file

- Cons:
  - Steep learning curve to read and write to these formats
  - Often requires extra software to be installed
  - More time required to use data across several tools (each tool/language has different API)

A Path Foreword: Data Format Transformation Service

It seems clear from the possible data format based solutions above that a combination of the pros from the *Standard Layout ASCII format* and the *standard, self-describing, machine-independent data format* would be ideal. However, both of these solutions have cons related to either the reading or the writing of each format. Therefore, we are proposing a solution in which the burden of writing to *a standard, self-describing, machine-independent data format* is removed from the data provider, and the pain of reading from a *standard, self-describing, machine-independent data format* is removed from the data user: a Data Format Transformation Service.

The Data Format Transformation Service should:

- Give data providers an easy front end 'wizard'-like interface to collect metadata and parsing information to transform their *User Defined ASCII format* into a *standard, self-describing, machine-independent data format* without any more effort than that needed to write a readme file (that is, the process would require no coding or special expertise on the data providers' part.)

- Give data users an easy way to retrieve the data in either a Standard Layout ASCII format file or a *standard, self-describing, machine-independent data format* file, as dictated by their own scientific workflow.

  This would enable data portals the ability to:

- Use the *Standard, self-describing, machine-independent data forma*t to serve data with rich services like aggregation and subsetting.
- Automatically extract metadata from the *Standard, self-describing, machine-independent data format* to create an automatically generated README-like file with a standard level of quality.
- Provide their users with access to a wide variety of datasets in a format that works for their scientific workflow, enhancing the reusability of data.

Benefits of Rosetta:

- *Rosetta* uses Unidata's Common Data Model and netCDF-Java I/O Service Provider (IOSP) layer, the latter of which provides the ability to read from and write to virtually any data format via the creation of an IOSP plugin. The flexibility afforded by the IOSP layer is that more transformation services can be added as the community requests them, with relative ease.

- Greater interoperability enabled by the choice of the netCDF format, which conforms to the CF conventions, as the *Standard, self-describing, machine-independent data format*.
- Offers a solution to part of the long-tail, "dark data" problem in which investigators face the challenge of sharing data that may not conform to any standard.
- *Rosetta* employs an open source development model and uses standard web technologies (Spring, JavaScript, Java).
- *Rosetta* uses transparent development infrastructure, such as github (for open source code management), and the use of the Jira ticketing system, which exposes the status of user submitted bugs, internal and external feature requests, and release timelines.
- Unidata Community, a diverse association of a colleges and universities, including instructors, researchers, and students. The NASA, NOAA, NSF, UCAR, and NCAR communities, as well as others, help inform Unidata, as these institutions are important sources of information and advice about events and conditions that can profoundly affect the program.

<u>Current state of Rosetta:</u>

- Conversion of simple datalogger output (in a *User Defined ASCII format* or spreadsheet format, such as .xls or .xlsx) into *Standard, self-describing, machine-independent data format* (netCDF) files that are compliant with the Climate and Forecast (CF) 1.6 standard specification. Efforts so far have been based on the needs of the Advanced Cooperative Arctic Data and Information Service (ACADIS) community.
- Limited test server online, with plans to open testing to a wider audience with the next release.

<u>Short term plans:</u>

- Expand the number of Discrete Sampling Geometries (DSGs) supported by *Rosetta* as specified in the CF-1.6 standard.
- Expand the number of input and output formats for conversion, including the output of a *Standard Layout ASCII format* and a spreadsheet based standard layout. (Standards for ASCII and spreadsheet layouts will be developed as part of a longer-term goal, described below.)
- Create an API, with guidance from the ACADIS group, for data publishing from *Rosetta* to various data portals.

Long term plans:

- Create and advocate for standard ASCII and spreadsheet representations of the CF-1.6 DSGs, based on the work described in the short-term plan above.
- Create a version of *Rosetta* suitable for local installation, for use with datasets that are too large to be efficiently transformed over the network.
- Expose THREDDS services, such as the netCDF Subset Service, through *Rosetta*, so that these services can be used on local files without running a full THREDDS Data server

Development of Rosetta in light of Unidata's stragitigic plan:

The Roestta group's work supports the following Unidata funding proposal focus areas:

1. **Enable widespread, efficient access to geoscience data**

   The initial goal of *Rosetta* is to transform unstructured ASCII data files into the netCDF format; once in this format, standard tools, such as the THREDDS Data Server, IDV, Python, and other analysis packages, can take advantage of these datasets with relative ease.

2. **Develop and provide open-source tools for effective use of geoscience data**

   Although the primary goal of *Rosetta* is to get data into the netCDF format, the transformation process does not stop there. The *Rosetta* group realizes that not everyone knows how to work with netCDF files, and may feel more comfortable working with other formats. Therefore, *Rosetta* includes the ability to transform from one format to another (e.g. netCDF to .xls), thereby reducing data friction.

3. **Provide cyberinfrastructure leadership in data discovery, access, and use**

   Metadata contained in netCDF format file (no longer locked away in a separate README file) can be automatically extracted, facilitating the discovery of data in these files. Additionally, the *Rosetta* development plan includes the creation of a standard ASCII and spreadsheet representations of the CF-1.6 DSGs.

4. **Build, support, and advocate for the diverse geoscience community**

   Promote the use of standard formats in the dissemination of data, while allowing flexibility to transform into other formats, as needed, to enable users to "do science." For commonly used formats, such as User Defined ASCII format or an unstructured spreadsheet, create and advocate for the use of a standard representations based on the CF-1.6 DSGs.