# EML - Bug #1794

## modify temporalCoverage to support ongoing data sources

12/01/2004 05:41 PM - Matt Jones

| | | | |
|---|---|---|---|
| **Status:** | In Progress | **Start date:** | 12/01/2004 |
| **Priority:** | Immediate | **Due date:** | |
| **Assignee:** | Matt Jones | **% Done:** | 0% |
| **Category:** | eml - general bugs | **Estimated time:** | 0.00 hour |
| **Target version:** | EML2.2.0 | **Spent time:** | 0.00 hour |
| **Bugzilla-Id:** | 1794 | | |

**Description**

---- Posted on behalf of Barbara Benson (bjbenson@wisc.edu) ----

I would like to raise some concerns that have arisen while developing EML documents for the North Temperate Lakes LTER.

Our data reside in an Oracle database, and tables are updated with new data at frequencies ranging from hourly to annually. We are creating EML documents to describe these data, and the data can be accessed dynamically from our website. Data from instrumented buoys are uploaded to the database every hour and are thus accessible from our website current to within the last hour. Our problem comes from trying to create temporal coverage for the NTL data. In order to have valid EML, it would seem like our options are:

1) to inaccurately describe the end date of a data set by choosing a static date; for example, the EML Best Practices document suggests using the end of the current year
2) to choose not to populate temporal coverage, thus having data sets that won't be located by temporal searches
3) to create data sets outside our database that are static
4) to use the "kluge" solution from a previous draft of the EML Best Practices using the alternative time scale as "ongoing" and leaving the end date blank.

For data sets that are only updated annually, we are willing to create an end date and just change that end date each year in the metadata. We have not decided how to handle temporal coverage for data that are updated more frequently but none of the currently available (valid) options seems desirable.

The current focus for creation of EML documents is to harvest them to the Metacat at the LTER Network Office. The rationale for this harvest is to support the data discovery functionality through Metacat across the LTER datasets. Given the well developed functionality of the NTL dynamic database access and the capability of capturing information about users accessing the NTL data, we want the EML documents to point to our dynamic database access system for each data set. Therefore, we don't find the creation of a static dataset a viable option at the present time when our higher level of functionality is not available centrally and not likely to become available in the near future.

To me the problems with creating temporal coverage for an ongoing data set highlight what I perceive to be a more general problem regarding the conceptualization of what objects EML is designed to describe. The set of objects needs to be bigger than static data sets. There are other data sources that need metadata description, e.g., database tables that are frequently updated, data streams from sensor networks. Some features of the current version of EML seem to be limited by this "static dataset" paradigm. It isn't hard to envision applications for EML attached to data streams.

We would appreciate your response to these issues. We think the next version of EML should accommodate ongoing data sets and allow the end date to be blank.

thanks

## History

#### #1 - 12/01/2004 05:46 PM - Matt Jones

We have discussed the problems associated with ongoing data. In particular, see the recent thread in eml-dev on this subject:

http://www.ecoinformatics.org/pipermail/eml-dev/2004-October/001027.html
http://www.ecoinformatics.org/pipermail/eml-dev/2004-October/001028.html
http://www.ecoinformatics.org/pipermail/eml-dev/2004-October/001029.html

So, we need to come to resolution on this issue.

#### #2 - 09/22/2008 12:20 PM - Margaret O'Brien

targeting for 2.1.0, although may drop back to unspecified.

#### #3 - 09/25/2008 02:43 PM - Matt Jones

Looks like the links to previous EML-dev discussion threads were wrong in Comment #1. The actual discussion occurred here:

http://mercury.nceas.ucsb.edu/ecoinformatics/pipermail/eml-dev/2004-October/001030.html
http://mercury.nceas.ucsb.edu/ecoinformatics/pipermail/eml-dev/2004-October/001031.html
http://mercury.nceas.ucsb.edu/ecoinformatics/pipermail/eml-dev/2004-October/001032.html

The synopsis of the discussion is this: I maintain that temporal coverage should refer to what data now exist (even in a dynamic database) and can be retrieved at the time the metadata are queried, not what the sampling protocol is. This allows queries to use the coverage information to accurately retrieve relevant data. Information about future intended sampling that has not yet occurred should go in sampling design descriptions, which will tell people what is intended but not make coverage inaccurate if plans change. In contrast, others feel that it is ok to have a 'null' field or to hack the field type and put in 'ongoing' or the like. My feeling is that doing this makes it indeterminate as to whether a search engine should return a data set for any given temporal search, and therefore reduces the search effectiveness of the metadata. For example, if someone has a so-called 'dynamic' database and enters a metadata record in 2002 saying that data span "2000- ", should a search engine return a 'hit' for a search for data in the range of 2008? What if the research project ended and those data weren't collected after 2004? Would we forever be obliged to return that data set as a hit, even for a search for data in 2020? To me this is an issue more about metadata accuracy and update frequency than anything else.

#### #4 - 10/01/2008 11:19 AM - Margaret O'Brien

Summary of 2008-09-30 discussion (Matt Jones, Margaret O'Brien, James Brunt, Mark Servilla, Inigo San Gil, Chris Jones, Corinna Gries, Ken Ramsey)

Consensus:
1. a resource-level endDate element with date content is necessary if accurate searches are to be returned
2. it is important that EML is able to accurately describe datasets in which values are expected to be added beyond the resource-level endDate (e.g., sensor data or time series)
3. authors wish to encode this "ongoing" nature in resource-level metadata, not just at the methods/sampling level

Comments and observations:
1. most sensor metadata will be autogenerated, and so keeping metadata up to date should not be a hardship. The frequency of update can be specified in the optional tag: /eml/dataset/maintenance/maintenanceUpdateFrequency (from appinfo/description: "Frequency with which changes and additions are made to the dataset").
2. Generally, it has been expected that searches of EML documents would be for quality-controlled products rather for unexamined real-time data.

The high-frequency metadata updates that would be required to keep a description of real-time data accurate may demand a different model for metadata storage applications (i.e, metacat), since repeated additions of metadata documents may overburden a system where the entire revised document is stored rather than just "diffs".

Possible changes to EML:
1. endDate element could have a optional attribute to indicate that data are expected be updated, e.g., <endDate updateFrequency="daily">
2. an element <metadataCreationDate> could be added. This need is also highlighted in bug #1991, which outlines other metadata maintenance issues.
3. add a new data type to described sensor data,  e.g., "dataStream", which would be analogous to dataTable
4. adapt the methods/sampling tree to accommodate an empty (or nillable) endDate (any changes to methods trees are related to bug #3504)

A subset of eml-dev members will consider these issues:
Ken Ramsey, Corinna Gries, Chris Jones, Matt Jones

#### #5 - 11/05/2008 03:17 PM - Margaret O'Brien

This bug is re-targeted for EML2.2
Reasoning: none of the simple solutions solve the basic issue -- that there is a difference between metadata written from the consumer and producer points of view, and that EML needs to accommodate both.
Probably the best solution will be to add another data type for streaming data, which could accommodate descriptions of existing ongoing collections, as well as work with emerging data streams like data turbine and sensorML.

#### #6 - 03/27/2013 02:18 PM - Redmine Admin

Original Bugzilla ID was 1794