Attributes and Units in LTER Data Packages (SBC-LTER)

2009-03-03, v 0.9, M. O’Brien

Includes Excerpts from and modifications of
EML Best Practices for Units DRAFT (LTER IMC Units Dictionary Working Group)
2009-02-13
and
EML 2.1.0 Schema Documentation (EML Development group)
2009-03-01

1. Introduction

In general use, the term ‘attribute’ defines a property of an object, element or a file (in computer science). In database vocabulary, a table column is called an ‘attribute’. If data were arranged in rows instead, then a row name could also be called an attribute. So in ecology and environmental sciences where data are often arranged in tables or arrays, attributes may be referred to as variables, parameters, columns or field names. A ‘unit’ is defined as “a particular physical quantity, defined and adopted by convention, with which other particular quantities of the same kind are compared to express their value.” (quoted from eml-docs, find a ref).

There is often a blending or overlap between units and attributes in local laboratory conventions. But on a structural level and for an unambiguous comparison of measurements, the attribute and unit must be distinguished. Units may be one of the most problematic categories of metadata. For instance, there are many attributes that clearly have no unit, such as named places and letter grades. There are other attributes for which a unit is difficult to identify, despite some suspicion that one should exist (e.g. pH, dates, times). In still other cases, a unit may be meaningful, but apparently absent due to dimensional analysis (e.g. grams of carbon per gram of soil). The relationship between units and dimensions likewise is not completely clear.

Anyone describing a data table for LTER (a dataset “author”) will need a moderate understanding of a) how to define the table’s attributes, b) when and how to define a unit, and c) the relationship between the two. This document is intended as a guide for those tasks. In our datasets (described in Ecological Metadata Language, ref here), a unit may be designated as either 'standard' or 'custom'. The list of ‘standard units’ was assembled to give dataset creators a collection of commonly used units to choose from. The EML developers recognized that it was not possible to include every conceivable unit, so there is a mechanism for including your own units as needed – which is why dataset authors need to understand the patterns. As a dataset author, you will probably notice that the units in the standard list are sometimes named inconsistently. Consequently this document’s secondary purpose it to promote consistent and proper unit construction in places where the EML schema and documentation fall short. It should be noted that although these recommendations have been derived from NIST standards (which is international), they are also consistent with the EML2.0 and 2.1 specifications, and so use English words instead of international symbols. The EML developers and the LTER IM Committee are aware of this shortcoming.

2. Attributes

A complete attribute description is composed of several parts. The guidelines here are to assist in choosing attribute names, measurement types (measurement scale), and their units (if appropriate).

2.1. Attributes Names and Labels

When naming and labeling table attributes (i.e., ‘variables’), any strings are allowed, including acronyms or ad hoc groups of letters. Column names like “temp_degC” or “daily NPP mgC” or any other abbreviations are fine, or even necessary to distinguish columns while keeping names short. Please avoid symbols since these can be misinterpreted by different computing systems; e.g., don’t use ‘@’ for ‘at’. A label is optional, but can be longer, capitalized, or use whole words to clarify the display on the web or in another application

2.2. Measurement Scale

When defining an attribute in EML, you will be specifying a “Measurement Scale”, which is a data typology borrowed from Statistics and introduced in the 1940's. Under the adopted model, attributes are classified as ‘nominal’, ‘ordinal’, ‘interval’, ‘ratio’, or ‘dateTime’ (EML Spec- ref). This classification is well-known although sometimes criticized. It provides at least first-order utility: for example, by allowing a unit to be included only for ‘interval’ and ‘ratio’ measurements EML prevents meaningless inclusion of units for categorical data, which should be classified as ‘nominal’ or ‘ordinal’.

The measurement scales range from simple to complex:
Nominal: values that can be considered categories. Values are assigned to distinguish them from other observations. Examples: using the number 1 for male and 2 for female, a species code or binomial, or the name of the site where the observation was made. Columns that contain strings or simple text are nominal type.
Ordinal: values are categories that have a logical or ordered relationship to one another, but the magnitude of the difference between values is irregular or is not defined. Examples: academic grades: A, B, C, D, F, or ranking quality 1=high, 2=medium, 3=low.
Interval: is used for data which consist of equidistant points on a scale, i.e., it is ordinal but the magnitude between the steps is known. Examples: the Celsius scale is an interval scale, since degrees are equally spaced but there is no natural zero point. Since the ‘0’ of the Celsius scale is tied to a property of water, 20 C is not twice as hot as 10 C. Another example is pH.
Ratio: is used for data which consists of equidistant points that also have a meaningful zero point, which allows ratios between values to have meaning. Examples of a ratio scale include the Kelvin temperature scale (200K is half as hot as 400K) and length in meters (e.g., 10 meters is twice as long as 5 meters). Concentrations are of ratio type.
dateTime: is for Gregorian dates and times, which have characteristics of both the ordinal type (ordered categories) and interval type (equidistant points on a scale). By making dateTime a separate category and providing an unambiguous mechanism for describing date formats, datasets contain the information needed to parse date values into their appropriate components (e.g., days, months, years).

2.3. Units

In EML, only measurement types **Interval** and **Ratio** include units. “The unit type ‘dimensionless’ is preserved,” and “is synonymous with ‘unitless’ and represents the case in which units cannot be associated with an attribute for some reason, despite the proper classification of that attribute as interval or ratio. Dimensionless may itself be an anomaly arising from the limitations of the … measurement scale ..”. (http://knb.ecoinformatics.org/software/eml/eml-2.0.1/index.html [5])"

2.3.1 Unit Type:

Morpho users will use its pull down menu. This table of examples was reproduced from SBC’s Morpho Users Guide (2009). [To Do: a better description of what a unit type is.]

example observations	**Unit Type**	**example units (standard)**
number, count	dimensionless	number
length	length	meter
biomass	mass	grams, kilograms
matlab datenumber, elapsed time	time	nominalDay, second
temperature	temperature	Celsius
density of kelp or benthic critter	areal density	numberPerMeterSquared
standing crop	areal mass density	gramsPerMeterSquared
phytoplankton density	volumetric density	numberPerMeterCubed
concentration, dissolved nutrients	Amount of substance concentration	molesPerLiter
particulate CHN	mass density	milligramsPerMeterCubed
water currents	speed	metersPerSecond
stream discharge	volumetricRate	litersPerSecond, cubicMetersPerDay
production rate	areal mass density rate	gramsPerMeterSquaredPerYear

2.3.2 Naming Units:

Although you have freedom naming attributes, you should keep in mind that the unit should be described in correct physical units. You may be tempted to create a custom unit that has “carbon” included because your lab always says “mg C per m2 per day” instead of using the EML standard unit “milligramsPerMeterSquaredPerDay”. But the proper construction is to use the unit’s standard form and also to make sure that you have included the element ‘carbon’ in the attribute’s name or description. You’ll notice that we write out the terms, rather than using abbreviations. This has the disadvantage of being long and English-centric, but it reduces the ambiguity of (e.g.) groups of slashes or parentheses, and does not introduce characters that are specialized or prone to misinterpretation. Other methods of describing attributes are being developed that will allow more advanced constructs, and which will allow important specific pieces of information (like an element) to be specified.

3. Examples and Common mistakes

These examples come from custom units that were submitted in EML documents to the LTER network. The Units Working group analyzed several hundred submissions and observed patterns in how authors were creating their custom units. The “proper” constructions also follow recommended practices for stating the base unit first, then its exponent (see below).

Pattern A. Naming conventions were not used. A custom unit was created when a standard unit is available, or a custom unit was created that did not follow naming conventions (LTER Unit Best Practices Draft 5.1).

Attribute	**Proper**	**Improper**
branch length	Meter	m
sampling area	metersSquared	squareMeters

Pattern B. part of the attribute was included in the unit name. The correct unit may have been already available as a standard unit (NIST#11; LTER Unit Best Practices Draft 5.2.2)

Attribute	**Proper**	**Improper**
bacterial abundance	Number	bacteria
primary production	milligramsPerMeterSquaredPerDay	mgC/m3/day
short shoot growth	numberPerCentimeterSquared	short shoots per cm2

Pattern C. A unit not required because the attribute was inappropriately typed. An attribute was given a measurement type "interval" or "ratio" but should have been dateTime, nominal or ordinal, which have no unit. Calendar dates and time durations are sometimes confused (see the dateTime type, above and section 5, below; LTER Unit Best Practices Draft, section 5.3)

Attribute	**example value**	**Proper**	**Improper**
date	2009-01-06	type: dateTime pattern: YYYY-MM-DD	type: ratio unit: yyyy-mm-dd
day of year	345 35.5	type: interval unit: nominalDay (standard)	type: ratio unit: day (custom)
datenum (in this case, from Matlab)	7668554.3455	type: interval unit: nominalDay (standard)	type: ratio unit: day (custom)

4. Recommendations for creating custom units

4.1. Check what others in the community are doing before creating a new unit.

4.2. Place only the most broad and essential measurement information in the unit, all other information belongs at the attribute level.

4.3 Some software produces a numerical representation of a time duration, which is simpler to use in calculations, or can be used internally to refer to a Gregorian date. When these are placed in datasets, their measurement type is interval (i.e., there is no real significance to a day ‘0’) and the unit is often nominalDay, that is, the difference between any 2 values is one day.

4.4. Naming Conventions

LTER IMC naming conventions for Custom Units were developed after observing inconsistencies in the standard units shipped with EML, and the confusion this occasionally causes. Recommendations for naming conventions for custom units in EML are based on NIST and SI recommendations.

4.4.1 Name first the unit then the modifier, i.e. 'meterSquared' rather than 'squareMeter'. This applies to each element in the unit name if multiple base units are brought together into a derived unit, ie. gramsPerMeterSquaredPerSecondSquared

4.4.2. Use ‘per’ as a linking term in the unit definition, ie, ‘gramsPerMeterSquared’.

Use camel case notation: the words are concatenated together, and all but the first are capitalized.

4.4.3. Spell out the unit rather than depending on abbreviations or shorthand. Use a spell checker (‘Microspell’ is free, and capable of finding misspellings in camelCase.).

4.4.4. Use only alphabetic characters in the unit name.

4.4.5 Numbers are allowed are in the unit abbreviation but not symbols or special characters.

4.4.6 Singular terms are preferred over plural terms (ie. gram vs. grams)

5. Gregorian Dates (excerpt from EML specification, URL #2)

Date/time values are from an interval scale, but are extremely complex because of the vagaries of the calendar (e.g., leap years, and leap seconds). The duration between date-time values in the future cannot be determined because leap seconds are based on current measurements of the earth's orbit. Consequently, date-time values are unlike any other measured values. The format string for date-time values allows one to accurately calculate the duration between two measured date-time values in the SI time parent unit ‘second’, assuming that the conversion software has a detailed knowledge of the Gregorian calendar. Note that this field would not be used if one is recording simple time durations. In that case, one should use a standard unit such as seconds, nominalMinute or nominalDay, or a customUnit that defines the unit in terms of its relationship to SI second.

Example(s):
YYYY-MM-DDThh:mm:ss
YYYY-MM-DD
YYYY
hh:mm:ss
hh:mm:ss.sss

5.1. Dates should be accompanied by a format string that conforms to the ISO 8601 standard using the following symbols.

Y	year
M	month
W	month abbreviation (e.g., JAN)
D	day
h	hour

m	minute
s	second
T	time designator (demarcates date and time parts of a dateTime)
Z	UTC designator, indicating value is in UTC time
.	indicates a decimal fraction of a unit
+/-	indicates a positive or negative number, or a positive or negative time zone adjustment relative to UTC
-	indicates a separator between date components
A/P	am or pm designator

5.2. Here are some examples of the format strings that can be constructed in the ISO 8601 standard

	Format string	**Example value**
ISO Date	YYYY-MM-DD	2002-10-14
ISO Datetime	YYYY-MM-DDThh:mm:ss	2002-10-14T09:13:45
ISO Time	hh:mm:ss	17:13:45
ISO Time	hh:mm:ss.sss	09:13:45.432
ISO Time	hh:mm.mm	09:13.42

5.3. Here are some examples of the format strings that are allowed, although they are not standard

Non-standard	DD/MM/YYYY	14/10/2002
Non-standard	MM/DD/YYYY	10/14/2002
Non-standard	MM/DD/YY	10/14/02
Non-standard	YYYY-WWW-DD	2002-OCT-14
Non-standard	YYYYWWWDD	2002OCT14
Non-standard	YYYY-MM-DD hh:mm:ss	2002-10-14 09:13:45

Some notes about these examples. First, the ISO 8601 standard is strict about the order of date components and the separators that are legal. Best practice is to follow the ISO 8601 format precisely. However, we recognize that existing data contain non-standard dates, and existing equipment (e.g., sensors) may still be producing non-standard dates. Consequently, EML allows formatting strings with additional characters to describe these date formats. In particular note that the use of a slash (/) to separate date components, a space to separate date and time components, using a twelve-hour time with am/pm designator, and placing any of the components out of descending order are all non-standard according to ISO. Nevertheless, these formats can be described using the format string to accommodate existing data.

Decimal date-time values can be extended by indicating in the format string that additional decimals may appear. Only the final unit (e.g., seconds in a time value) can use the extended digits according to the ISO 8601 standard. For example, to show indicate that seconds are represented to the nearest 1/1000 of a second, the format string would be "hh:mm:ss.sss". Note that this only indicates the number of decimals used to record the value and not the precision of the measurement (see dateTimePrecision for that).

6. Checklist for units

The National Institute of Standard and Technology (NIST) provides a reminder of the uncertainty involved in units as well and rules in the form of a checklist http://www.physics.nist.gov/cuu/Units/checklist.html [9]. Several of the points on this checklist address specific issues that have arisen in recent LTER community discussions of units and are listed here for easy reference. This checklist id appropriate for any resource containing units, not just LTER datasets (e.g., published papers).

NIST #2, #8, #18 Abbreviations: Abbreviations such as sec, cc, or mps are avoided and only standard unit symbols, prefix symbols, unit names, and prefix names are used. The combinations of letters "ppm," "ppb," and "ppt," and the terms part per million, part per billion, and part per trillion, and the like, are not used to express the values of quantities. Acronyms or ad hoc groups of letters should not be used as units.

NIST #3 Plurals: Unit symbols are unaltered in the plural (e.g., use 75 cm, not 75 cms.)

NIST #5 Multiplication & division: A space or half-high dot is used to signify the multiplication of units. A solidus (i.e., slash), horizontal line, or negative exponent is used to signify the division of units. The solidus must not be repeated on the same line unless parentheses are used.

NIST #9 Unit modifications: Unit symbols (or names) are not modified by the addition of subscripts or other information. For example, use Vmax = 1000 V, not V= 1000 Vmax

NIST #10 Percent: the symbol % is used to represent simply the number 0.01 proper: “D = 0.2 %, where D is defined by the relation D = (l1 - l2)/l2”. Improper: “the length l1 exceeds the length l2 by 0.2 %”

NIST #11 Information & units: Information is not mixed with unit symbols or names. Proper: “the water content is 20 mL/kg”. Improper: “20 mL H2O/ kg”

NIST #12 Math notation: It is clear to which unit symbol a numerical value belongs and which mathematical operation applies to the value of a quantity.

proper	35 cm x 48 cm	20 °C to 30 °C	123 g ± 2 g or (123 ± 2) g	240 x (1 ± 10 %) V
improper	35 x 48 cm	20 °C-30 °C	123 ± 2 g	240 V ± 10 % (one cannot add 240 V and 10 %)

NIST #13 Unit symbols & names: Unit symbols and unit names are not mixed and mathematical operations are not applied to unit names.

proper:	kg/m3	kg · m-3	kilogram per cubic meter	
improper	kg/cubic meter	kg per m3	kilogram per meter3	kilogram/m3

.

NIST #22 Obsolete Terms: The obsolete terms normality, molarity, and molal and their symbols N, M, and m are not used.

proper	concentration of B and its symbol cB and SI unit mol/m3 (or a related acceptable unit)	molality of solute B and its symbol bB, or mB and SI unit mol/kg, (or a related unit of the SI)
improper	normality and the symbol N	molarity and the symbol M, molal and the symbol m

7. EML representation of created units UNDER CONSTRUCTION

This section for those editing XML documents, and not for those using metadata management tools such as Morpho.

Ref 2004 Best practices doc (http://location.in.the.doc.archive)

Murray-Rust and Rzepa (ref) state for scientific units “It is likely that several groups will develop a variety of approaches. STMML supports units through dictionaries so that is easy for a community to create new units. Sufficient information is held to manage dimension analysis, and to support the "dimensionless" unit in a richer manner.” Units are represented in EML using STMML tags <unitList> and <unit>. Units have an associated , that represents a set of units with a name. Units are described with the name and description tags

string describing the unit

The relationship of the unit to SI (parent, multiplier) must be designated if there is to be support for unit conversions.

9. **References**

Attachment	Size
Taylor, ed, 1995. Guide for the Use of the International System of Units (SI). (sp811.pdf) [16]	412.71 KB
Taylor, ed, 2001. The International System of Units (SI). (sp330.pdf) [17]	813.13 KB
Murray-Rust and Rzepa, 2002. STMML. A Markup Language for Scientific... (ds121.pdf) [18]	239.98 KB
List of standardUnits in CVS format: standardUnits_EML201.txt [19]	10.34 KB
SI Unit rules and style conventions, checklist for reviewing manuscripts [9]	
EML Best Practices for LTER Sites, Oct 2004 (WRONG LINK)	

Source URL #1: http://intranet.lternet.edu/im/news/committees/working_groups/unit_dictionary/EML_bestpractices_units_DRAFT
Source URL #2: http://knb.ecoinformatics.org/software/eml/

Links:
[1] http://intranet.lternet.edu/im/news/committees/working_groups/unit_dictionary/EML_bestpractices_units_DRAFT
[2] http://www.bipm.org/en/si/
[3] http://en.wikipedia.org/wiki/ISO_31
[4] http://www.ch.ic.ac.uk/rzepa/codata2/
[5] http://knb.ecoinformatics.org/software/eml/eml-2.0.1/index.html
[6] http://intranet.lternet.edu/im/im_practices/metadata/guides
[7] http://www.physics.nist.gov/cuu/index.html;
[8] http://physics.nist.gov/cuu/Units/bibliography.html
[9] http://www.physics.nist.gov/cuu/Units/checklist.html
[10] http://www.bipm.org/en/si/si_brochure/
[11] http://www.jstage.jst.go.jp/article/dsj/1/0/1_128/_article
[12] http://cvs.ecoinformatics.org/cvs/cvsweb.cgi/eml/eml-unitDictionary.xml
[13] http://water.usu.edu/cuahsi/odm/cv.aspx
[14] http://his.cuahsi.org;
[15] http://river.sdsc.edu/Wiki/Default.aspx
[16] http://intranet.lternet.edu/im/files/im/sp811.pdf
[17] http://intranet.lternet.edu/im/files/im/sp330.pdf
[18] http://intranet.lternet.edu/im/files/im/ds121.pdf
[19] http://intranet.lternet.edu/im/files/im/standardUnits_EML201.txt