EML Handbook

Version 2: Working Draft

Written by:

David Blankman
Jeanine McGann
LTER Network Office
University of New Mexico
Albuquerque, NM
Rick Reeves
Mark Schildhauer
NCEAS
© 2004

Table of Contents

Chapter Page Contents

4.3 Description: Individual EML M

<Mark S writing here>

1.0 Forward

1.1 Objective

The EML Handbook is a comprehensive collection of conceptual descriptions, detailed technical information, and case-study tutorials describing the design and implementation of Ecological Metadata Language (EML). This document assumes a basic level of computer knowledge, and some experience working with ecological data.

1.2 Intended Audience

This document is designed for ecologists, ecological information managers, and ecology students as a comprehensive reference to the Ecological Metadata Language, EML. It describes EML's overall structure, and presents tutorial material in the form of case studies, that demonstrate EML's use in creating ecological ***Data Packages***. For those seeking a more technical or comprehensive overview of EML and its capabilities, this document provides links to several online electronic documents. These are listed at the end of Section 1.3.

1.3 Guide to Chapters

Chapter 2, ***EML Overview,*** presents a general introduction to EML. Senior scientists, data managers, and others seeking a mid-to high-level understanding of the way in which EML is used in the scientific workplace should read this chapter first.

Chapter 3, ***Case Studies: Creating and Maintaining EML,*** depicts the EML production process using three interrelated case studies. This section includes a demonstration of different approaches to creating and editing EML. Ecologists and research assistants charged with entering and maintaining EML should read this chapter first.

Chapter 4, ***In-Depth: Creating Data Packages Using EML*** presents extensive technical details for EML, focusing on techniques construction of metadata Data Packages using EML modules. The chapter is intended for EML users who require detailed understanding of the EML specification.

Three Appendices contain additional EML-related technical information: ***Appendix 1,*** contains the complete EML Specification, including schema diagrams depicting the EML architecture. ***Appendix 2,*** Sample Ecological Data, presents the sample ecological data tables (and accompanying metadata) used in the Chapter 3 case studies. ***Appendix 3, The EML Implementation Guidebook,*** is a stand-alone, introductory guide to hosting an EML-based metadata repository at your research site.

The following online electronic documents provide additional information regarding the design and use of EML. Several EML Handbook chapters refer readers to these documents:

- **knb.ecoinformatics.org/software/eml**: Complete EML Specification

- **knb.ecoinformatics.org/software/morpho**: Morpho Data Management Tool

2.0 EML Overview

2.1 High-level Definitions: Data Package, Metadata, and EML

2.1.1 The Data Package: A System for Documenting Research Datasets

Ecologists collect data describing many aspects of their data collection efforts. This information is often combined with the actual experiment data into an encapsulated collection; this collection is subsequently stored in a searchable database where it is potentially available to scientific community. This collection is known as a ***Data Package.*** Note that at this point the Data Package is a language-and device-independent concept. To be useful it requires implementation via tangible rules, language, and creation-storage-retrieval technology.

2.1.2 Metadata: Systematic Description of Datasets

Simply put, metadata is information that describes, in a systematic manner, other information. ***The Dublin Core Metadata Initiative (1)*** Web site provides a comprehensive definition from within the context of Library Science:

> *Metadata has been with us since the first librarian made a list of the items on a shelf of handwritten scrolls. The term "meta" comes from a Greek word that denotes "alongside, with, after, next." More recent Latin and English usage would employ "meta" to denote something transcendental, or beyond nature. Metadata, then, can be thought of as data about other data. It is the Internet-age term for information that librarians traditionally have put into catalogs, and it most commonly refers to descriptive information about Web resources.*
>
> *A metadata record consists of a set of attributes, or elements, necessary to describe the resource in question. For example, a metadata system common in libraries -- the library catalog -- contains a set of metadata records with elements that describe a book or other library item: author, title, date of creation or publication, subject coverage, and the call number specifying location of the item on the shelf.*
>
> *The linkage between a metadata record and the resource it describes may take one of two forms: elements may be contained in a record separate from the item, as in the case of the library's catalog record; or the metadata may be embedded in the resource itself.*

Also consider this functional metadata description, presented through an example: Members of the ecological community compile descriptions of their experimental observations in many ways: they may take “mental notes”, intending to transcribe them

upon return to their office. They may accumulate written notes in the margins of field notebooks, interspersed within the actual data, or in separate files. These descriptions: who, what where, when, why and how an ecological dataset was collected, are ***metadata***. Without accompanying metadata, a scientific data set becomes more difficult to use, and therefore less valuable to scientists, with the passage of time.

One could say that ***any*** analysis that utilizes independently collected data is not possible without metadata. For example, consider a data table that is presented without metadata describing the columns and rows. Such a data table is useless as the numbers have no scale, units, or context. Further, the if there is no Point of Contact attached to the data set, there is no way to contact the entity responsible for collecting it, thus no way to acquire these key parameters.

Metadata may be implementated within an organization in several ways; however, to maximize its usefulness it is best to adopt a standardized, systematic approach, such as a ***metadata language***. These define both the content, format, and possibly, the implementation tools needed to create, implement, and search, and use ecological metadata. ***Ecological Metadata Language (EML)*** is an emerging metadata language that is gaining support within the ecological community.

2.1.3 EML: A Framework for Constructing Data Packages

Ecological Metadata Language (EML) is a metadata content specification *for ecological data*, resulting from a community-based effort involving ecological researchers, information managers, and software developers from around the United States, in particular at Long Term Ecological Research sites, LTER, and the National Center for Ecological Analysis and Synthesis, NCEAS. The need for EML has long been recognized within the ecological community as critical to the preservation and long-term intelligibility of the growing archives of ecological data (FLED Report 1995, Michener et al, 1997). EML was also designed to assist the individual researcher in managing and more effectively using their own data. EML attempts to be non-ambiguous and comprehensive, and is intended for use by any ecologist or ecological information manager who wishes to document their data in a way that facilitates its reusability and sharing.

EML describes a number of critical aspects of the data, such as variables' names and definitions, their units of measurement, time and place of data collection, who collected the data, etc. Much of this information is currently collected by ecologists in various forms—such as field notes, or marginalia, and hence are subject to being lost, separated from the data, or stored in very arbitrary ways. These disadvantages are eliminated by using a more formal and standard framework such as EML. Even in the case of existing data catalogs, the terminology and types of metadata collected vary widely, whereas *EML provides a standardized set of terms and definitions intended specifically for ecological metadata.*

EML is implemented using ***Extensible Markup Language*** (XML), a growing standard for marking

up documents on the Web. This means that metadata created in EML will be highly available for searching and exchanging over the Web, if so desired. This formal structure for EML not only provides a *common structure* for ecologists to document and interpret ecological data, but also one that better enables efficient development of software applications that process the metadata. The kinds of technology applications anticipated range from basic tools that allow users to perform specific targeted searches for datasets, to advanced applications that can perform automatic integration of datasets. This will be a tremendous help to ecologists in organizing and implementing long-term research studies, where the ability to search for and synthesize large amounts of data is crucial.

EML is a set of pre-defined terms, known as "tags" or "elements", and a set of rules. Each tag consists of a pair of tokens. The tag names enumerate different metadata categories; the actual metadata are placed between the tokens. The EML rules specify the attributes and behavior of each tag; for example, acceptable content, or the maximum permissible number of instances for each tag can be used for a particular data set, etc.

For example, the tag called ***abstract*** tag set must contain a "brief overview of the resource or data being documented". There should be one ***abstract*** tag set within each metadata package, and it should contain character/textual information. The value of the abstract tag for a particular data set might be:

> *"This dataset contains demographic information on the survival and reproduction of barbed goat grass (Aegilops triuncialis) populations reciprocally transplanted between core and edge sub-populations along an invasion front located at the Mclaughlin reserve. The purpose of the study is to examine the potential for rapid evolutionary change along invasion fronts of this exotic grass. The study will probably end in Summer, 2003."*

Notice that this use of "abstract" matches with the popular conception of the definition of an abstract. EML attempts to use familiar terms with intuitively corresponding definitions whenever possible. However, the need for consistency and reduction of ambiguity has often required adoption of more technically accurate terms for EML metadata tags, even when some other term might be widely used by practicing ecologists. For example term "attribute" refers to what ecologists frequently call a "variable" -- referring to data of a given type contained within a column of a data table. This naming choice corresponds with formal use in database terminology. Other alternative names for "attribute" might have included "field" or "column header". Faced with such choices, EML typically adopts the most technically appropriate choice.

To illuminate the process of designing metadata elements to store specific data fields in EML-based data packages, consider representing the *creator* of an ecological dataset within the data package. The EML Specification implements the creator element as a subset of the root-level ***eml-resource*** module, which (among other entities) stores general descriptive information about dataset

resources. Specific creator data fields are stored using the ***responsibleParty*** complex data type. The responsibleParty data type in turn uses other complex data types for example, the ***Person*** type, to store the attributes of the creator within the metadata element. The Person type in turn is comprised of (among other fields) fundamental character string and numeric elements; moving up the schema hierarchy, the eml-resource module uses several responsibleParty elements to contain other resource-specific attributes (e.g., the provider of the data package's metadata).

In summary, EML consists of:

- A set of XML-derived tags, comprised of nested hierarchies of data entities, that map to standard ecological data or metadata components;
- A set of rules establishing the relationship between and use of these tags.

Software tools, for example, the KNB project's ***Morpho*** metadata editor, simplify the EML creation process by hiding some of the formal complexities of the XML-based tag set. This is similar to the way that a Web browser such as Netscape encapsulates the complexity of the HTML standard to make Web pages more accessible.

Chapter 4 discusses the design and internal structure of EML in greater detail.

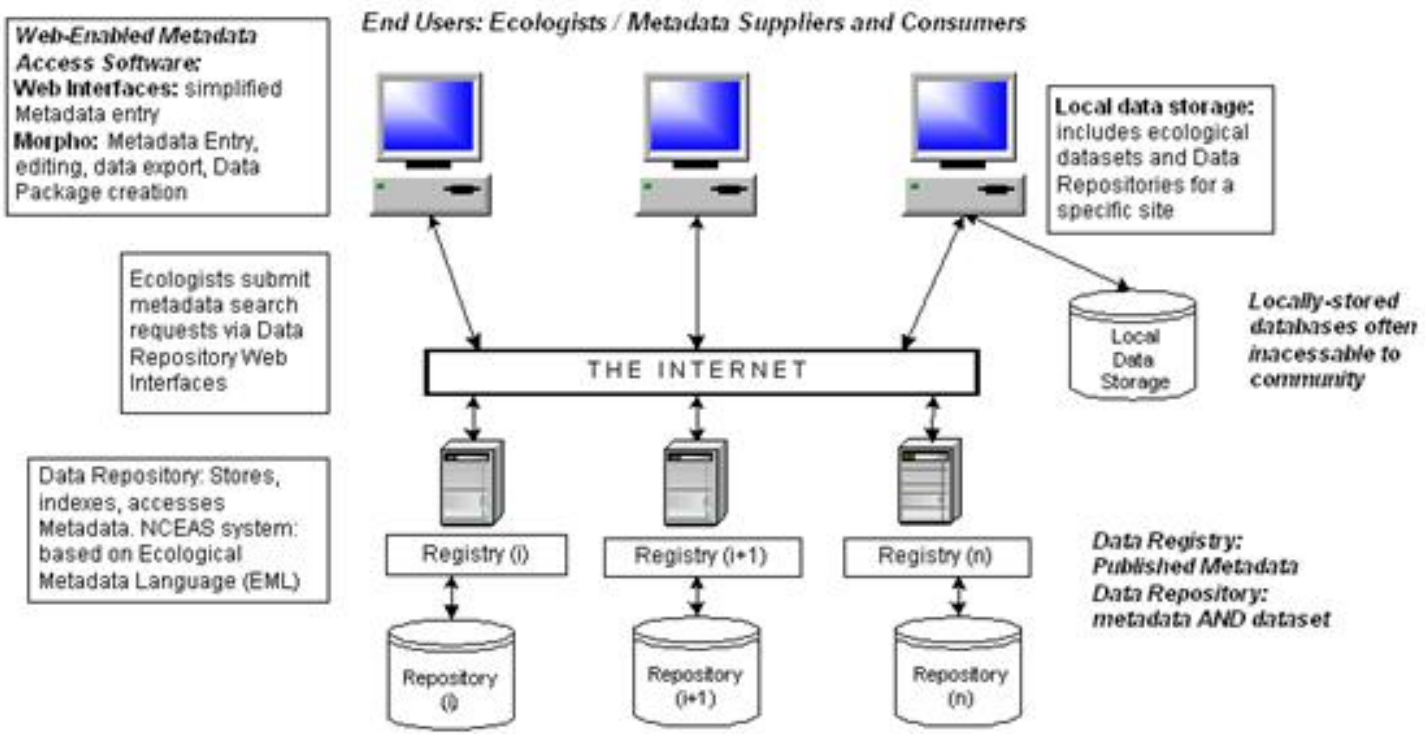
2.2 The Data Package Life Cycle

Data packages are living documents. They are created using as much metadata as exists at the time describing the dataset, when ecologists are motivated to document new datasets before metadata becomes stale. At some future point, new metadata can become available; for example, other research team members may return to base and contribute their observations. Or, the dataset originally described may be amended with new measurements that require further description. Finally, the DP may be transferred, or uploaded, into a physically separate second data repository.

Scientists employ desktop-based data management software to enter detailed descriptions of their research datasets (i.e., metadata) in a standardized format. If they desire, the researcher can also combine the actual research data sets with the metadata, forming a Data Package. The new Data Package is stored on the researcher's computer disk; At the researcher's option the package can also be exported to one or more Data Repositories maintained on remote computers and accessible via the Internet and special software. Two examples of such software are: 1) a Data Repository web site serving as a 'front-end' to a Data Repository; 2) specialized data management software that supports search, retrieval, and maintenance of data packages stored in one or more remote Data Repositories.

Figure 2.2 depicts the Data Package development and use process:

Figure 2.2: Data Package Production / Storage / Distribution



2.3 Setting the Stage: A Simple Metadata Example

To start the discussion of Data Package development using EML, we present a simple metadata/EML example: You have just completed a research project on rodent populations at a northwestern New Mexico field station. You are interested in creating EML-compliant metadata to document your research so that it can be loaded onto the station's EML metadata catalog. These metadata will enable other investigators to become aware of your work through Internet searches, while also serving as a durable "cataloguing" of the data set for your own convenience. Prior to creating your data, you ask the station's EML custodian: "What is the minimum set of metadata that constitutes a valid EML document?" The custodian presents you with a sample EML document (**Figure 2**), that shows the smallest valid EML document, which consists of a dataset tag, which in turn contains tags for a title, creator, and contact. Note the nesting of tags, such that some might contain actual metadata values (e.g. title), while others contain further tags that have values (e.g., creator contains the tag for **individualName**, which consists of tags for **surName** and **givenName**).

Figure 2
Example: minimal valid EML Document
[mps1]

```
<?xml version="1.0"encoding="UTF-8"?>
```

```
<eml:eml>
packageId="eml.1.1"system="knb"
xmlns:eml="eml://ecoinformatics.org/eml-2.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ds="eml://ecoinformatics.org/dataset-2.0.0"
xsi:schemaLocation="eml://ecoinformatics.org/eml-2.0.0 eml.xsd">
<dataset>
<title>Rodents of the Southwest</title>
<creator>
<individualName>
<surName>Ecosis</surName>
</individualName>
</creator>
<contact>
<individualName>
<surName>Ecosis</surName>
</individualName>
</contact>
</dataset>
</eml:eml>
```

[mps2] **All XML documents start with this. Most XML documents use UTF-8 also known as Unicode xmlns is short for XML Namespace. These fields are the minimum necessary to produce a valid EML dataset document: <title> <creator> <contact> must be unique for the SYSTEM** [mps3] [mps4]

You can see that this EML document doesn't contain much information—just some basic contact information and indication of content, if the title is well composed. Your new question is: what additional elements might provide sufficient information to enable other scientists to determine whether sharing or combining your data would lead to a productive collaboration? The remainder of this document endeavors to answer this question; it will begin by examining in greater detail the structure of the sample EML document.

2.3.1 [mps5] *Documenting an ecological dataset*

The fundamental "object" for which an ecologist might want to create metadata is termed a "**dataset**" in EML. "**dataset**" means different things to different people. In EML, the term may contain general information such as the title, creator, and contacts, as well as one or more data entities, such as data tables, that provide more specific research details. Take a look at the expanded EML dataset schema displayed in Figure 3. This schema provides a graphical representation or "map" of the actual XML schema that shows where the different elements fit in the hierarchy of the language. **[mps6]**

For now, consider only the ResourceGroup schema, which is displayed in **Figure 4**. The

<title> field provides a description of the resource being documented that is long enough to differentiate it from other similar resources. Multiple titles may be provided, particularly when trying to express the title in more than one language. If a title is in a language other than English, use the “xml:lang” attribute. For example,

<title>**Rodents of the Southwest**</title>
<title>xml:lang=”de”**Rodents des Südwestens**</title>

In the second <title>, the text ‘xml:lang=“de”’ is an XML attribute that says: language =german (de=deutsche). There are two versions of ISO 639 language codes: two- or three-letter codes. The three-letter system was designed to expand the list of languages that can be represented. Currently, the two-letter system is used more often, but over time it is likely that the three-letter system will be embraced.

A clear discussion of the use of the language attribute can be found at: http://msdn.microsoft.com/library/default.asp?url=/library/en-s/prompting_beta2/html/SSML_Elements_lang.asp.

The actual language codes can be found at: http://www.w3.org/WAI/ER/IG/ert/iso639.htm.

[mps7] [mps8] A <creator> is a structure for representing the owner of the data. A <creator> can be a person, an organization, or an organizational role. Examples of organizational roles are: Executive Director, Department Administrator, or Information Manager.

At this point, we expand the minimal EML document to look at the <creator> information in more detail.
[mps9]

...<creator>
<individualName>
<salutation>**Dr.**</salutation>
<givenName>**Pat**</givenName>
<givenName>**Morgan**</givenName>
<surName>**Ecosis**</surName>
</individualName>
<address>l
<deliveryPoint>**Department of Southwestern Ecology**</deliveryPoint>
<deliveryPoint>**Semiarid State University**</deliveryPoint>
<city>**Noaguaville**</city>
<administrativeArea>**NM**</administrativeArea>
<postalCode>**78890**</postalCode>
<country>**USA**</country>
</address>
<phone>**709-345-8970 x 254**</phone>
<electronicMailAddress>**pat.ecosis@semiarid.edu**</electronicMailAddress>
</creator>

```
<creator>
<organizationName>USDA</organizationName>
<address>
<deliveryPoint>U.S. Department of Agriculture</deliveryPoint>
<deliveryPoint>Mailstop 3654</deliveryPoint>
<deliveryPoint>25 Federal Plaza</deliveryPoint>
<city>Washington</city>
<administrativeArea>DC</administrativeArea>
<postalCode>20025</postalCode>
<country>USA</country>
</address>
<phone>phonetype=”voice”709-345-8970 x 254</phone>
<phone>phonetype=”fax”709-345-8962</phone>
<electronicMailAddress>dataservices@usda.gov</electronicMailAddress>
<onlineUrl>http://www.usda.gov/ecoinformatics/</onlineUrl>
</creator>
```

The element names may seem unfamiliar and you might be wondering why the developers of EML used such strange names. These elements have been taken from the International Standards Organization (ISO), schema for representing people: iso-party. For example, a person can have more than one <givenName>, but only one <surname>. EML does not have a specific tag for middle names. Since <givenName> is optional, <surName> would be used for “Cher” or “Madonna.”

An address can have more than one <deliveryPoint>. A <deliveryPoint> is that part of an address that precedes the <city>. The most common <deliveryPoint> is a street address. Other examples are company names, department names, or post office boxes. Because XML does not recognize carriage returns, the way to enter that part of an address that precedes the city is to use a separate <deliveryPoint> element for each item that you would want to appear on a separate line when displayed on a web page or printed report. An <administrativeArea> is the tag that would be used in the United States to represent a state or in Canada to represent a province. [mps10]

A <creator> can have more than one address. Each address must be contained in its own set of <address></address> tags:

```
…<creator>
<individualName>
<salutation>Dr.</salutationr>
<givenName>Joe</givenName>
<surName>Ecologist</surName>
</individualName>
<address>
<deliveryPoint>25 Oceans Avenue</deliveryPoint>
<city>Oceans</city>
<administrativeArea>CA</administrativeArea>
<postalCode>98025</postalCode>
</address>
<address>
<deliveryPoint>Department of Ecological Sciences</deliveryPoint>
<deliveryPoint>University of the Oceans</deliveryPoint>
```

<city>**Oceans**</city>
<administrativeArea>**CA**</administrativeArea>
<postalCode>**98025**</postalCode>
</address>
</creator>

In terms of EML schema, the above code fragment would look something like this:

There are two additional optional creator types that can be described by EML, <metadataProvider> and <associatedParty>. A <metadataProvider> can be used if the person, organization, or organizational position that provided the metadata is not the one who created the data being documented. A <metadataProvider> could be an LTER information manager, a student, or a researcher who may or may not have been part of the data creation process. While this field is optional, it is recommended that it be used if the metadata was provided by someone other than the <creator>.

It is also a good idea to use this [mps11] if there are multiple creators, but one of them is the primary source of the metadata. It is especially important to use this structure if the metadata is being constructed after the fact. For example, if someone decides to create EML documentation for data that is 20 years old, some metadata may exist in a lab notebook, but the documenter also is developing new metadata.

An <associatedParty> is a person, organization, or organizational position that is involved in the creation of the data, but is neither a <creator> nor a <metadataProvider>. An associated party could be a researcher, statistician, technician, or an advisor or consultant. If the research data was developed by a team of researchers, the principal investigator might be the <creator>, while her associates could be described by <associatedParty>. The schemas for both <metadataProvider> and <associatedParty> have essentially the same structure as that of <creator>, however, it should be noted that <associatedParty> also contains a mandatory element called <role>, which is used to define the role that the party played in relation to the research being documented, such as “research assistant,” or “technician.”

As mentioned earlier, the structure of an EML document is defined by a series of XML schemas. The schema determines not only the names of valid tags, but also the order of the various structures. As can be seen in Figure 4, a <creator> must be described before a <metadataProvider>, and a <metadataProvider> before an <associatedParty>. [mps12]

Aside from names and addresses, the ResourceGroup schema also is used to provide an overview of the information in the dataset. Some of the more useful ResourceGroup elements include <abstract>, <keywordSet>, <distribution>, and <coverage>. While they are not required elements, both <abstract> and <keywordSet> are useful for describing the general nature of the research being documented, especially when the data is being

designed for use with a search engine. The <distribution> and <coverage> elements reappear in later modules, and as they are also optional, they will be discussed later. All of these elements, however, should be listed after the <creator>, <metadataProvider>, and <associatedParty>.

In the <abstract> section, each paragraph is separated by the <para></para> tagset, while each section is enclosed in the <section></section> tag set. For most abstracts, there will be only one section.

As for keywords, the entire set is defined by the <keywordSet> tags, while each individual keyword is separated by a <keyword></keyword> tag set.

```
    …<abstract>
    <section>
<para>Small mammal species are often affected by changes in vegetation composition and structure. In the southwestern United States, many large recent burns have altered the structure and function of piñon ecosystems. These burns have affected the structure of the dead wood habitats of many mouse species. In this study, small mammals in a northern New Mexico piñon and juniper forest were live-trapped in a non-burn area and it was found that mouse species (especially Peromyscus species) were associated with many dead wood habitats.
</para>
</section>
</abstract>
<keywordSet>
<keyword>pinon forest</keyword>
<keyword>Peromyscus maniculatus</keyword>
<keyword>deer mouse</keyword>
<keyword>New Mexico</keyword>
</keywordSet>
```

The next element to be added after the abstract and keywords is the contact information. A <contact> is a structure for representing the person, organization, or organizational role to contact regarding the use of the data. [mps13] Although <contact> followed <creator> in the minimal EML document in Figure 2, a closer look at **Figure 3** and **Figure 4** reveals that <creator>, <metadataProvider>, and <associatedParty> are part of the larger structure of the ResourceGroup, while <contact> is located just below the ResourceGroup in the schema. Any elements that are used in the ResourceGroup must be entered before the next set of elements <purpose> (optional), <maintenance> (optional), and <contact> (required).

Without getting too deep into the language of XML schema, we can say that both the <contact> and <creator> elements have the same structure, that is, both are “of type ResponsibleParty.” The <contact> information follows the same structure as the <creator> information with regard to address and phone and online resources.

```
…<contact>
<positionName>Information Manager</positionName>
```

```
<address>
<deliveryPoint>Oceans LTER</deliveryPoint>
<deliveryPoint>Department of Marine Ecology</deliveryPoint>
<deliveryPoint>University of the Oceans</deliveryPoint>
<deliveryPoint>1514 San Ysidro</deliveryPoint>
<city>Seaside</city>
<administrativeArea>LA</administrativeArea>
<postalCode>78890</postalCode>
<country>USA</country>
</address>
<phone>709-345-8970 x 254</phone>
<electronicMailAddress>imanager@oceans.edu</electronicMailAddress>
<onlineUrl>http://oceanslter.lternet.edu</onlineUrl>
</contact>
```

One other element that should be discussed relevant to the entire dataset is <access>. This element specifies permissions given to certain groups or individuals regarding access to the data or metadata contained in a particular file. Within <access>, there is a choice of <allow> or <deny>, both of which then require both <principal> and <permission> elements. The <principal> refers to the name of the individual or organization to which the rule applies, while <permission> specifies the level of access that is being granted, either "read," "write," "changePermission," or "all."

```
…<access>
<allow>
<principal>Sevilleta LTER</principal>
<permission>read</permission>
<permission>write</permission>
</allow>
</access>
```

2.3.2 Introducing the data entity

We have now defined some of the basic elements of EML pertaining to identifying the who, what, when, and where of the data. Clearly it is beneficial to create some additional metadata that describes the structure and contents of the data itself. As stated earlier, a dataset consists of one or more data entities, and the most common data entity is a <dataTable>. A data table is a very familiar construct to most ecologists—and usually looks something like this:

Table 1

OBSERVATION_DATE	OBSERVATION_LOCATION	SPECIES	SPECIES_COUNT
2002-10-29	FCE_001	ALLIGATOR	1
2002-10-29	FCE_002	ALLIGATOR	2
2002-10-29	FCE_003	ALLIGATOR	0

Data tables are produced by people using software applications such as text or word processors and they often are saved as text files (ascii) or spreadsheets (Excel), by statistical packages (SAS, SYSTAT, SPSS), by database systems (MS Access, My SQL, Oracle, MS SQL Server), or by certain kinds of sensors (data loggers).

In addition to data tables, people using database applications may also produce a <view> from a database management system or a <storedProcedure> that results in data output.

People using GIS (geographical information system) applications generate both <spatialVector>, also referred to as boundary or shape files, and <spatialRaster>. A <spatialRaster> is a geo-referenced image usually produced by a camera on a satellite or other remote sensing device.

The final kind of data entity is <otherEntity>. An <otherEntity> is a data entity that cannot be represented by any of the previously defined data entity structures. A non-geo-referenced photograph is an <otherEntity>, e.g. a photograph of two different types of butterflies.

The elements must be entered in the sequence shown in Figure 6, and any element that was specified in the ResourceGroup applies to the entire dataset that is being documented. This is especially significant if the dataset has more than one data entity. For example, assume the dataset consists of three files: Climate.xls, Biodiversity.xls, and Productivity.xls. In this case, ResourceGroup elements such as <distribution> or <coverage> must be general enough to apply to all of the files. This would be true also if Climate.xls, Biodiversity.xls, and Productivity.xls were the names of tables in a database.

2.4 EML Document Production and Management Tools

2.4.1 Creating EML with Text Editor

It is possible, but not advisable, for users to create a data package using a text editor. To do so, user creates an editable ASCII text file containing the ‘raw’ metadata, and then embeds the appropriate EML tags in the correct places throughout the metadata to create a valid EML document. Two main drawbacks: 1) producing a valid EML document entirely ‘by hand’, without supporting software to validate the EML syntax, is time-consuming, error-prone work. 2) There is no way to embed non-ascii data (e.g. a binary image or database file), in metadata constructed in a text editor. Therefore, the authors do not recommend this approach except for extremely simple metadata cases, or case where the researcher has absolutely no other choice.

2.4.2 Using Web-Based Data Registry

Some institutions and research projects are creating Web enabled data entry forms that collect key metadata fields relevant to the kinds of data sets produced by their affiliated researchers. These forms are then presented to the research community as a Data Registry web site. Data entered into these Data Registries are stored in project-specific Data Repositories. The advantage of these sites is they focus the researcher’s efforts on collecting and entering he metadata required for that project. The disadvantage is that these Data Registries typically do not support editing of existing data packages, or entry of any ‘off-project’ metadata fields.

2.4.3 Using Specialized Metadata Editing Tool

Specialized metadata entry tools provide the data package developer with complete data entry and edit capability for any valid EML field. One such tool is the **Morpho** data managent tool, described in the next section

> ***2.4.3.1 The Morpho Data Management Tool:*** Morpho, designed to enable ecologists to create EML-based data packages. It provides an easy-to-use, cross-platform application that accesses and manipulates metadata and the accompanying data sets, both locally and network-wide. Morpho allows ecologists to create both standardized, metadata descriptions and to create catalogs of data and metadata upon which to query, edit, and view data collections. Consult the **Morpho User's Guide** for further information; various Morpho features are demonstrated in **Section 3.0.**

3.0 Case Studies: *Morpho*-based Data Packages

3.1 *Morpho* Data Package Wizard

The Morpho data management tool is comprehensive, and thus complex for novice metadata developers to use. To support researchers needing to create data packages, and without requiring them to master Morpho's advanced editing capabilities, this tool includes the Data Package Wizard. The Wizard is a series of interactive data entry forms that collect the most critical and commonly used metadata / EML fields from users to produce a valid data package. The Morpho User Guide (link here) demonstrates use of the Data Package Wizard.

In the following three sections, we will extend the tutorial example presented in the User Guide to demonstrate data package creation for three progressively comprehensive metadata entry cases. The data used in these examples are found in the EML Guidebook (link here).

3.2 Case 1: Package with Data Tables ONLY

Coming Soon

3.3 Case 2: Data Tables with Limited Metadata

Coming Soon

3.3 Case 3: Data Tables with Comprehensive Metadata

Coming Soon

4.0 In Depth: Understanding and Using EML

The objective of this chapter is to connect the advanced metadata developer with advanced technical

information concerning EML. Specifically, this includes a discussion of EML's modular design and references the formal **EML Specification**. The information in this chapter is of interest to persons designing advanced EML-based data packages, or to anyone requiring advanced, detailed knowledge of EML architecture. After studying the material in this chapter, metadata developers will be able to parse the EML Module descriptions in the EML Specification, select the modules required to construct a Data Package

4.1 EML Design Overview

The Ecological Metadata Language (EML) is a metadata standard developed by the ecology discipline and for the ecology discipline. It is based on prior work done by the Ecological Society of America and associated efforts (Michener et al., 1997, Ecological Applications). The purpose of EML Is To provide the ecological community with an extensible, flexible, metadata standard for use in data analysis and archiving that will allow automated machine processing, searching and retrieval.

EML is comprised of modules that, in concert, provide a comprehensive ecological metadata solution. This modularization is intended to assist information managers in selecting only those aspects of metadata which are of importance to their organizational needs, as well as to provide for easier flexibility and integration with other possible metadata standards. The modular design of EML will usually be hidden from the individual scientists who are using EML.

Still, it is useful to understand that these modules do exist in EML, and are designed to encompass broad areas of thematic interest. Each module addresses general issues, such as 1) basic descriptions about data set ownership—who, what, where, when, and why; or 2) descriptions about how the data are actually structured; or 3) what are the distribution rights for the data? For example, below we list some general metadata concepts along with the appropriate EML tags:

1. Ownership

   Who did the research? **(creator)**
   In general terms, what is the research about? **(abstract)**
   What are some of the key concepts addressed by the data? **(keywords)**
   Where was the research done? **(geographic coverage)**
   What time periods are covered in the data? **(temporal coverage)**
   What species are represented in this research/data? **(taxonomic coverage)**
   What methods were used? **(methods)**
   Is the data being documented part of a larger project **(project)**

2. Data architecture

   The data representation branch contains modules that are used to describe:
   What kind of data entity is it? **(e.g., Table vs. Raster)**
   What are the names of variables within the data set? **(attribute name)**

3. Distributional information

How would I retrieve this file? **(physical)/(distribution)**
Who will I allow access to the data? **(access)**

These three examples list a subset of EML's available modules and tags **The essence of creating (meta)data packages using EML is to understand which tag to use for which type of metadata, and then to properly fill in the tags which are critical to your (and others) needs. In addition, as a researcher, you might benefit by reviewing the overall EML set of tags, as you might discover critical types of metadata worth documenting that you hadn't previously considered.**

Following sections of this document describe EML in more detail, presenting specific examples intended to familiarize ecologists with the meaning and appropriate use of many common types of ecological metadata.

4.2 EML Design Objectives

EML is implemented as a series of XML document types that are used in a modular and extensible manner to document ecological data. Each EML module is designed to describe one logical part of the total metadata that should be included with any ecological dataset.

The architecture of EML was designed to serve the needs of the ecological community, and has benefited from previous work in other related metadata languages. EML has adopted the strengths of many of these languages, but also addresses a number of short-comings that have proved to inhibit the automated processing and integration of dataset resources via their metadata.

Four key design features permeate the EML implementation:

- **Modularity:** EML was designed as a collection of modules rather than one large standard to facilitate future growth of the language in both breadth and depth. By implementing EML with an extensible architecture, groups may choose which of the core modules are pertinent to describing their data, literature, and software resources. Also, if EML falls short in a particular area, it may be extended by creating a new module that describes the resource (e.g. a detailed soils metadata profile that extends eml-dataset). The intent is to provide a common set of core modules for information exchange, but to allow for future customizations of the language without the need of going through a lengthy 'approval' process.

- **Detailed Structure:** EML strives to balance the tradeoff of too much detail with enough detail to enable advanced services in terms of processing data

through the parsing of accompanied metadata. Therefore, one key question driving EML design was: 'Will this particular piece of information be machine-processed, just human readable, or both?' Information was then broken down into more highly structured elements when the answer involved machine processing.

- **Compatibility:** EML adopts much of it's syntax from the other metadata standards that have evolved from the expertise of groups in other disciplines. Whenever possible, EML adopted entire trees of information in order to facilitate conversion of EML documents into other metadata languages. EML was designed with the following standards in mind: Dublin Core Metadata Initiative, the Content Standard for Digital Geospatial Metadata (CSDGM from the US geological Survey's Federal Geographic Data Committee (FGDC)), the Biological Profile of the CSDGM (from the National Biological Information Infrastructure), the International Standards Organization's Geographic Information Standard (ISO 19115), the ISO 8601 Date and Time Standard, the OpenGIS Consortiums's Geography Markup Language (GML), the Scientific, Technical, and Medical Markup Language (STMML), and the Extensible Scientific Interchange Language (XSIL).

- **Strong Typing:** EML is implemented in an Extensible Markup Language (XML) known as XML Schema, which is a language that defines the rules that govern the EML syntax. XML Schema is an internet recommendation from the World Wide Web Consortium, and so a metadata document that is said to comply with the syntax of EML will structurally meets the criteria defined in the XML Schema documents for EML. Over and above the structure (what elements can be nested within others, cardinality, etc.), XML Schema provides the ability to use strong data typing within elements. This allows for finer validation of the contents of the element, not just its structure. For instance, an element may be of type 'date', and so the value that is inserted in the field will be checked against XML Schema's definition of a date. Traditionally, XML documents (including previous versions of EML) have been validated against Document Type Definitions (DTDs), which do not provide a means to employ strong validation on field values through typing.

Two concepts clarify the underlying EML design: the ***content model*** (i.e. the concepts behind the structure of a document - which fields go where, cardinality, etc.), and the ***syntactic implementation*** of that model (the technology used to express the concepts defined in the content model). The normative sections below define the content model and the XML Schema documents distributed with EML define the syntactic implementation. For the foreseeable future, XML Schema will be the syntactic specification, although it may change later.

4.3 Description of Individual EML Modules <*Mark S writing here*>

EML is comprised of 22 structural elements known as ***modules:***

- eml **(Coming Soon...)**
- eml-access
- eml-attribute
- eml-constraint
- eml-coverage
- eml-dataset
- eml-dataTable
- eml-entity
- eml-literature
- eml-methods
- eml-party
- eml-physical
- eml-project
- eml-protocol
- eml-resource
- eml-software
- eml-spatialRaster
- eml-spatialReference
- eml-spatialVector
- eml-storedProcedure
- eml-text
- eml-view

Each module implements a separate syntactic element; a (meta)data package will be comprised of one or more of these elements. Sometimes more than one instance of each element will occur in a metadata package. Some elements are hierarchical in that they contain other elements as members. Thus are elements combined in many unique ways to create a potentially infinite combination of data packages. Each module is described below, the **Appendix** to this report provides additional technical information.

[mps1]Maybe what you could do is move this raw XML formatting to the Appendix, so that an interested Data Manager could see each example in its raw format. I don't think it belongs out here-- it is difficult to read, and obscures the main purpose, which is to familiarize scientists with the content of these various EML metadata tags.

[mps2]Move to appendix. Too technical

[mps3]

I [mps4]don't understand what is SYSTEM?

[mps5]Suggest move ALL of the Anatomy Section to a technical appendix.

[mps6]Again, I think it worthwhile simplifying these diagrams to make them more understandable.

[mps7]Nice concise statements like this are very valuable. The notion to get across is—what is a "creator"—what elements/tags are used to define a "creator", etc.

[mps8]The general approach in this section is quite good! I have not provided detailed comments, since so much hinges on NOT using the XMLSPY figures, and NOT showing the raw XML. The explanatory text, however, reads very well. Please keep in mind to focus on covering the MOST important or generally useful tags, to the extent that this is predictable. Your choices seemed generally good, although, for example, I don't think the language example deserves mention so early. Move that to a technical appendix.

[mps9]I think even in examples you should try to use some real sounding data, and drop the attempts at humor. Also, keep examples sounding more out of a typical fields station or individual ecology researcher. Examples can easily assist someone starting out if they are close enough to an actual situation—lilke a grad student or faculty person who has just accomplished some field work at an organized field station.

[mps10]All of these tags and their definitions should be included in a glossary to this document for quick refernce!

[mps11]What is "this"?

[mps12]The notion of ordering might be something you need to say a word or two about earlier, especially if, as I highly recommend, you move most of the more technical stuff to an appendix. Similarly you'll need to describe in very simple terms the concept of reusable resources such as ResourceGroup and ResponsibleParty.

[mps13]I think you need to be able to describe this stuff without referring to a schema. Maybe the approach is to say—Whenever in EML you need to describe a "person"—there is a standard set of tags that must be used. We call this set of tags "ResponsibleParty", etc. You should put this generic description of reusable types right after your lead-in example of a minimal EML document.