

出國報告（出國類別：研究）

研習生態資訊學以促進與建立台灣資訊系統研究網

服務機關：行政院農業委員會林業試驗所

姓名職稱：陸聲山 助理研究員

派赴國家：美國

出國期間：93 年 9 月 26 日～93 年 12 月 23 日

報告日期：94 年 1 月 20 日

系統識別號：C09400449

行政院及所屬各機關出國報告提要

頁數： 53 含附件：是

報告名稱：研習生態資訊學以促進與建立台灣資訊系統研究網

主辦機關：

行政院農業委員會林業試驗所

聯絡人/電話：

郭麗娜/23039978 # 1118

出國人員：

陸聲山 行政院農業委員會林業試驗所 助理研究員

出國類別：研究

出國地點：美國

出國期間：民國 93 年 9 月 26 日-民國 93 年 12 月 23 日

報告日期：民國 94 年 1 月 20 日

分類號/目：F8/林業 F8/林業

關鍵詞：長期生態，後設資料(metadata)， EML，Morpho

內容摘要：

生態研究在長期的資料蒐集過程中，已累積了數量龐大且差異性極大的資料檔案，加上資料檔案分散各地不易取得，使得即時獲取資料的分享機能無法發揮，不僅難以擴大資料的應用面，更因許多資料庫缺乏管理極易造成資料的流失，不但減損了這些研究資料的價值，同時也相對提高了研究成本，更無法整合既有資料，產生新的研究價值。因此如何在龐雜、分散與異質性的資料中找到特定需要的資料；如何確保資料的長久保存與取得；以及如何確保資料的分享與整合，一直是我們生態研究人員所面臨的難題。為了解決這些問題，需要建立一套標準，使得各系統之間有所遵循，才能進行資料的存取、索引、檢索與共享，後設資料(metadata)便是扮演這樣的角色與功能，而生態後設資料語言(Ecological Metadata Language, EML)便是一種根據生態學科而發展出的後設資料規格。本報告將介紹經由美國長期生態研究網(LTER Network)及美國國家科學基金會(NSF)等共同贊助與發展的生物複雜性知識網路(Knowledge Network for Biocomplexity, KNB)計畫項下所開發出的 EML，作為生態資料庫聯結與整合的共同標準，建立符合 EML 規格的後設資料，將有助於我們日後對資料的存取與利用，並促進資料的分享與整合。

本文電子檔已上傳至出國報告資訊網

目 次

| | |
|-------------|----|
| 一、目的----- | 1 |
| 二、過程----- | 2 |
| 三、研習內容----- | 3 |
| 四、心得----- | 13 |
| 五、建議事項----- | 14 |
| 六、附錄----- | 15 |

一、目的

傳統在生態學與生物多樣性的研究上，研究者經常是對各自所蒐集到的資料進行分析，而且所獲得的資料往往僅由一位或少數幾位研究者進行較小規模或尺度的觀察和試驗，因此難以推導出共同或通用的生態理論。相對的，現今生態上和環境上的許多重大議題，例如全球暖化問題、生物多樣性的消失、外來種的入侵或自然資源的耗竭等研究，越來越趨向要瞭解在大規模的空間和時間尺度下的複雜生態系統，所以需要的資料已超過傳統蒐集單一資料的範圍，也越來越迫切要求資料的共享、整合，甚至能進行資料探勘(data mining)。這種改變也導致了生態學研究社群意識到必須增加對資料的分享與取得的能力，事實上許多觀察到的生態現象所發展出的理論，往往受限於能否取得適當的長期資料，而這也是長期生態研究值得被重視的原因之一。

而生態研究在長期的資料蒐集過程中，已累積了數量龐大且差異性極大的資料檔案，加上資料檔案分散各地不易取得，使得即時獲取資料的分享機能無法發揮，不僅難以擴大資料的應用面，更因許多資料庫缺乏管理極易造成資料的流失，不但減損了這些研究資料的價值，同時也相對提高了研究成本，更無法整合既有資料，產生新的研究價值。因此如何在龐雜、分散與異質性的資料中找到特定需要的資料；如何確保資料的長久保存與取得；以及如何確保資料的分享與整合，一直是我們生態研究人員所面臨的難題。為了解決這些問題，需要建立一套標準，使得各系統之間有所遵循，才能進行資料的存取、索引、檢索與共享，後設資料(metadata，即在一個標準化的格式下描述所蒐集到的資料)便是扮演這樣的角色與功能，因此後設資料的推動與實施是非常重要的。生態後設資料語言(Ecological Metadata Language, EML)便是一種根據生態學科而發展出的後設資料規格。

本研究計畫派遣具生態學背景之種子人員前往美國威斯康辛大學麥迪遜Madison的湖泊研究中心(Center for Limnology)，經由實際的參與計畫及實做中學習相關資訊技術，重點之一便在學習EML及其編輯工具Morpho等新技術之觀念及其應用。本報告將介紹經由美國長期生態研究網(LTER Network)、加州大學聖塔芭芭拉(Santa Barbara)分校的國立生態學分析與整合中心(NCEAS)、加州大學聖地牙哥分校之高速電腦中心(SDSC)、德州科技大學(Texas Tech

University)及美國國家科學基金會(NSF)等共同贊助與發展的生物複雜性知識網路(Knowledge Network for Biocomplexity, KNB)計畫項下所開發出的EML，作為生態資料庫聯結與整合的共同標準。首先將從EML的發展與目標、規格與特點、模組的組成與結構、EML的實施與各級標準做一概述、最後介紹一種容易使用的EML編輯工具Morpho，能使我們快速地建立符合EML規格的後設資料，將有助於我們日後對資料的存取與利用，並促進資料的分享與整合。

二、過程

| 日 期 | 起 訖 地 點 | 工 作 任 務 |
|------------|------------|---------|
| 9/26 (日) | 台北→洛杉磯/芝加哥 | 去程轉機 |
| 9/28 (二) | 芝加哥→麥迪遜 | 去程 |
| 9/28-12/21 | 麥迪遜 | 研習期間 |
| 12/21 (二) | 麥迪遜→芝加哥 | 返程轉機 |
| 12/23 (四) | 芝加哥→洛杉磯→台北 | 返程 |

三、研習內容

(一)、生態資料檢索標準 EML 之介紹

1. EML 的發展與目標

生態資料的長期價值及其效益，若要能促進生態學研究的進展和解決重要的環境問題，將取決於是否有適當和充分的後設資料，也就是要具有敘述資料內容、內涵、品質、結構和可及性等的敘述性資訊。EML 便是因應生態學的需求，根據生態學科而發展出來的後設資料規格，主要目的就是為了應用在生態學上。

早期的 EML 開發工作主要是由美國生態協會(Ecological Society of America)和相關附屬協會的努力而墊下基礎的，接著 EML 便成爲一個開放原始碼、社群導向的計畫，主要致力於提供描述與生態學資料有關的一個優質的後設資料規格，以促進有效的生態資訊管理。從最早 1997 年的 EML 1.0.0 版問世後，中間經過多次的版本修正，至 2000 年時已修正至 1.4 版的 EML，2002 年大幅修正後的 2.0.0 版 EML 正式上路，今年(2004) 2.0.1 版的 EML 又做了小幅的修正。在 EML 發展的過程中，包括了許多熱心的資訊技術人員貢獻了無數寶貴的時間和經驗，同時也包含了生態學家自身的參與，尤其是資訊管理人員。如此經過三方面人員的協同合作，才使得 EML 能符合與解決生態學科長久以來所面臨的資料分散、異質性、整合不易的問題。

實施 EML 規格的主要目標，使得資料與後設資料文件化有遵循的標準，是一種方便人與電腦皆可讀懂的語言，因此生態學家能有所遵循，利用共同的結構與標準提供所蒐集的資料，以便其他的生態學家也能正確地解讀其資料。此外 EML 的規格制定後，也促使程式撰寫人員得以開發出容易的操作介面或相關資料分析等應用軟體，大幅增進生態研究人員利用相關工具經營管理或分析他們的資料。更進一步的，透過資料上網註冊的機制，除確保資料的長久保存，亦促進資料的發現與取得、分享與整合，大幅擴大了生態學資料的長期利用性。

2. EML 的規格與特點

目前以 EML 為發展生態資料與後設資料的基礎架構有何好處呢？因為在 EML 的發展中，擷取了之前其它有關後設資料語言中的許多優點，但也發現其中一些應用在生態學上的缺失，尤其是關於資料集資源(dataset resource)自動化處理與整合的問題，也因此 EML 的逐步發展才能設計出滿足生態學研究社群需求的後設資料。可擴展式標記語言 (eXtensible Markup Language, XML) 是一種容許使用者定義標籤的標記語言，近年來已經成為資料儲存的標準格式，也逐漸的被視為資料交換的標準格式。EML 是一種符合 XML 文件標準的結構，參照 XML Schema 來描述生態研究上的後設資料，具有相當強的能力來描述複雜的資料關係，而且是一種文字型態的檔案，更方便日後資料的交換和維護。EML 的使用就如同 XML 文件型式一樣，採用一系列的模組(module)方式來將資料文件化，且具有擴展性(extensible)，能隨使用者的新增需求項目而擴充的一種方式。每個 EML 模組設計用來描述全部後設資料的一部分，而這整個後設資料文件應該涵蓋有任何一個生態學資料集。

EML 發展的目的是提供生態學研究社群以一種可擴充式的、靈活的後設資料標準，這種標準才使得電腦能自動化的處理、搜尋和檢索儲存的資料並進行分析和歸檔的工作。後設資料規格建立後，也就是描述資料的規格，是生態學資料分析和建立模型的基本要件。下面列出一些 EML 特點：

(1). 模組化(modularity)：EML 的設計發展是以不同模組的集合來代替單一龐大的標準架構，因此才能因應與促進 EML 未來在廣度和深度上的成長。經由 EML 以延伸性架構的實施，首先選擇一些模組做為核心模組，核心模組可適當的描述所具備的資料、文獻和相關軟體等資源。並且，如果 EML 在某一個特殊領域上沒有合適或相關的模組可利用，便可經由產生一個新的模組來描述該領域，亦即擴大 EML 的資料集。這種設計提供了共同核心模組的集合以做為為資訊交換之用，

但也考慮到語言本身的成長，允許未來客制化自行產生新的模組而不需要冗長的驗證過程。

(2). 詳細的結構(detailed structure)：資料內容越詳盡越好，才不會日後發現資料並無法解讀而導致資料價值降低甚至喪失。因此 EML 的設計中，也花了許多功夫在過多的細節或足夠的細節中要取得平衡點，使得我們根據處理資料(通過解析伴隨的後設資料)便能產生進一步的服務。所以，在 EML 設計過程中產生的一個問題便是：哪些資訊的特殊部分能被機器處理？或人可讀懂？或兩者皆可呢？因此所謂的資訊便被劃分為高度結構化的元素(element)，使得機器能進一步處理，以產生能因應我們需求的回覆。

(3). 相容性(compatibility)：EML 採取的大部分句法，其實也是參照了其它學科專門發展後設資料標準的技術而來，甚至儘可能的話 EML 採取其它後設資料整個資訊的樹形圖，如此便大幅增加 EML 文件与其它後設資料語言之間的轉換與相容。EML 的設計主要參考了有下列的標準：Dublin Core Metadata Initiative、Content Standard for Digital Geospatial Metadata (CSDGM)、Biological Profile of the CSDGM、Geographic Information Standard (ISO 19115)、ISO 8601 Date and Time Standard、Geography Markup Language (GML)、Scientific, Technical, and Medical Markup Language (STMML)、Extensible Scientific Interchange Language (XSIL)等。

(4). 強制型態(strong typing)：EML 的實施是以 XML 為架構，而 XML 架構是一個符合網際網路的共通標準，因此符合 EML 句法的後設資料文件結構，自然也就符合 XML 架構的定義標準。在 XML 架構下，使用強制資料型態(strong data typing)的方式，任何變數只能儲存於特定宣告過的資料型態中，因此提供了更好的資料檢驗能力。例如，若輸入變數為日期(date)的資料時，這些輸入的數值將根據 XML 架構中的日期定義再次被檢驗。

3. EML 的模組組成與結構

模組的組成是以嚴謹方式組織而成的邏輯單位，模組設計成為可重複使用，且具有自我描述的能力，資料本身被內嵌在 EML 文件化的檔案裏。eml 模組(eml module)就像一個包裝容器(wrapper container)一樣，在一份單一 EML 文件中包含容納了任何的後設資料內容，以一定結構的方式描述這些不同類型的生態學資源。在 EML 中稱為資源，其實也是根據都伯林核心後設資料的定義而來，主要用來描述一般被敘述為數位化資源(digital resource)的元素集。然而，它也允許使用描述非數位化(non-digital resource)的資源，譬如尚未被數位化的地圖或其他文物等。EML 上層結構的設計是與都伯林核心句法相容。

最上層的 eml 模組就如同容器(container)一般，但必須依靠次一層的模組(sub-module)進一步加以詳細的說明與描述，但自身提供了基本的鑑別(identification)資訊。上層模組，即 eml 資源模組(eml-resource module)，包含所有資源的基本資訊。主要有四個基本的資源類型(resource type)可供利用，也就是資料集資源(dataset resource)、文獻資源(literature resource)、協議資源(protocol resource)和軟體資源(software resource)等四個類型。各資源類型的描述雖有其特殊性，但之間仍具有一些共通的欄位，例如標題、創始者等。所有 EML 文件必須以<eml>標記作為開始，並在結束的地方以</eml>標記。有些模組可單獨使用，但一些模組必須依靠其他的模組才能完成整個的敘述。EML 透過從上而下的方式，先從上層模組開始將生態資料文件化，進一步的細節與描述將透過陸續增加的模組而完成。但是，有些模組在不同的階層上都可使用，例如存取模組(eml-access module)，只要適合時這類的模組便可被加以描述。

接下來有六個支持的模組(supporting module)，用來增加詳細的細節以說明上層的資源。六個模組主要包含以下的使用：描述存取控制資源(後設資料和資料)的規則(eml-access module)、資源的分佈情形(eml-physical module)、與資源相關的人或機構(eml-party module)、資源的時間、空間與分類學相關的

資訊(eml - coverage module)、關於資源與整體研究計畫相關的資訊(eml - project module)、關於資源方法上的資訊(eml - methods module)。支持模組增加每種資源更加充分的資訊內容，提供越詳細的資訊將可促進資料發現、解釋、和品質的控管等。

資料集(dataset)組織模組提供在邏輯結構上的許多細節，對機器進行資料處理非常重要。實體(entity)，對應於資料表(dataTable)，需要更詳細的資料描述，才能提供所需細節使機器可讀懂，得以進行資料集的處理，此外提供每個資料類型獨特的語法。空間參考(spatialReference)用來定義坐標系統，做為資料集對地球的空間座標參考。單位字典(unitDictionary)為資料集屬性提供單位的定義，方便進行不同單位係數之間的轉換運算。文字(text)模組是 eml 模組內的一個通用模組，為許多領域提供格式化標記，例如許多段落的結構都是利用文字方式來做敘述說明。eml 模組還有一個擴展性的部分叫做附加的後設資料(additionalMetadata)，以因應容納新增的資訊，允許任何的 XML 領域新增至 eml 模組文件中。

4. EML 的實施與各級標準

EML 的規劃與實施，將是生態資料分享與整合最基礎的資訊建設工作。雖然我們也逐漸體認到，資料長期的價值將取決於我們完善的後設資料紀錄與管理方式，但這種符合 EML 規格的資料歸檔工作，卻是非常耗費時間與精力的工作，也將是我們面臨資訊整合的最大瓶頸。所以美國長期生態研究網辦公室(LNO)組成了評估小組，以了解各研究站實施 EML 所面臨的各種問題，必要時並派員到該站提供所需要的協助，以加速各研究站將資料 EML 化。由 LNO 與 LTER 研究站的資訊管理員所評論過的 LTER 研究站 EML 最佳實施法(EML Best Practices for LTER Sites)，提出具體的建議與詳細描述 EML 各級完成所需的標準，使 LTER EML 文件共用性達到最大、異質性減到最低以促進資料的整合，提供各 LTER 研究站指

導原則，協助完成 EML 初步的實施，並訂定目標逐步改善 EML 的後續實施而達到完全符合 EML 文件化的最終功能。

EML 的各級標準，也就是各階段內容的完整性(completeness)，主要分成六級：第一級為資料鑑別階段(Identification level)、第二級資料發現階段(Discovery level)、第三級資料評估階段(Evaluation level)、第四級資料存取階段(Access level)、第五級資料整合階段(Integration level)、第六級語意用途階段(Semantic use level)。下表中列出各層級主要包含的元素及必要描述，各層級除了包含前一階段的元素外，並新增更多的元素加以描述，以提供資料資源更完整的後設資料描述，也因此能支援更多的功能，而達到最終資料整合的目標。

| Completeness Level | Description and Major Elements Added |
|--------------------------|--|
| 1: Identification | Minimum content for adequate data set discovery in a general cataloging system or repository (functionally equivalent to LTER DTOC): <ul style="list-style-type: none">• title• creator• contact• publisher• pubDate• keywords• abstract (recommended)• dataset/distribution (i.e. url for general dataset information) |
| 2: Discovery | Level 1 content, plus coverage information to support targeted searches, adding elements: <ul style="list-style-type: none">• geographicCoverage• taxonomicCoverage• temporalCoverage |
| 3: Evaluation | Level 2 content, plus data set details to enable end-user evaluation of the methodology and data entities, adding elements: <ul style="list-style-type: none">• Intellectual Rights• project |

| | |
|------------------------|---|
| | <ul style="list-style-type: none"> • methods • dataTable/entityGroup • dataTable/attributes (see issues outlined in the text) |
| 4: Access | Level 3 content plus data access details to support automated data retrieval, adding elements: <ul style="list-style-type: none"> • access • physical |
| 5: Integration | Level 4 content plus complete attribute and quality control details to support computer-assisted data integration and re-sampling, adding elements: <ul style="list-style-type: none"> • attributeList (full descriptions) • constraint • qualityControl |
| 6: Semantic Use | Level 5 content plus semantic information (currently under development by SEEK, and may require extension to the EML schema) |

生物複雜性知識網路(KNB)爲了協助生態學家能有一個容易使用與操作的介面，即使在不了解 EML 的規格、架構、各級標準等概念下，也能輕易的產生出一份 EML 文件，而開發出了 Morpho。Morpho 是一種容易使用且跨平臺的資料管理應用工具，使得生態學家能產生、獲取和操作 EML 後設資料和資料，並可查詢、觀看和檢索公開的生態學資料。接下來就 EML 產生工具 Morpho 做一簡介。

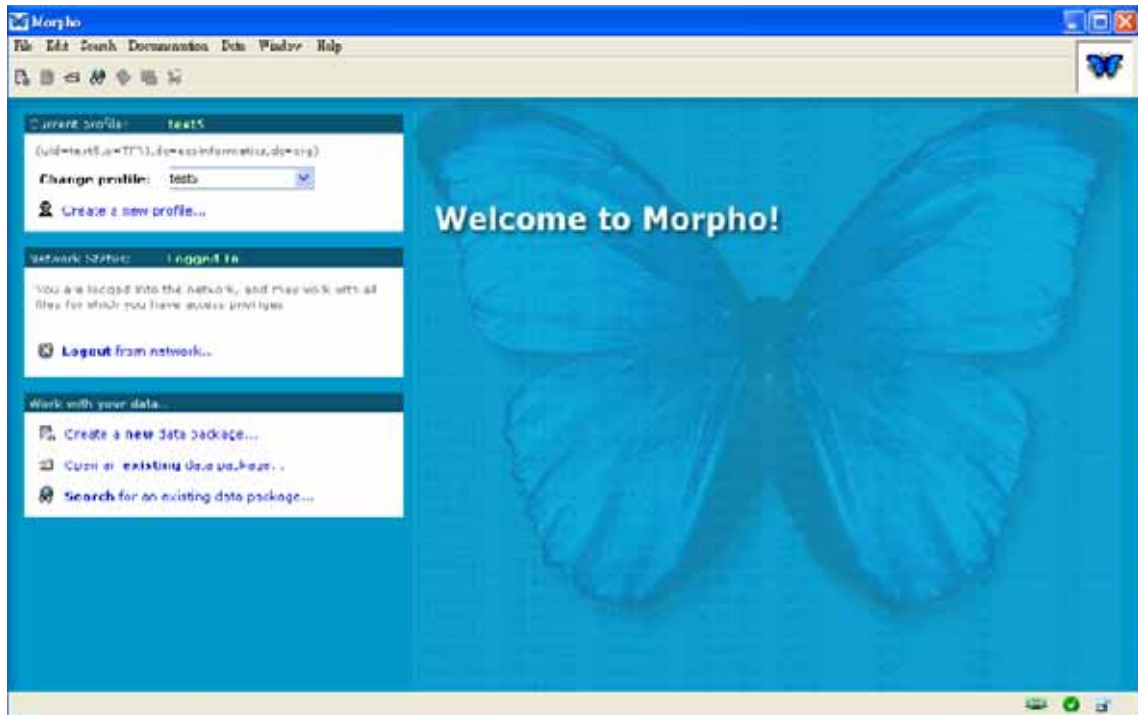
(二)、EML 產生工具 Morpho 簡介

Morpho 是專爲生態學家設計的一個資料管理工具，它提供了一個容易使用與跨平台應用的介面，使用者得以在本機和網路上取得和操作後設資料及資料。Morpho 使得生態學家可以輕易的產生符合 EML 標準的後設資料，並且也建立後設資料與相關資料的編目(catalog)，因此能促進查詢(query)、編輯(edit)和觀看(view)這些蒐集到的資料。另外，Morpho 也提供了對網路伺服器 Metacat 的存取功能，因此能對已公開的生態學資料進行相關的查詢、觀看和檢索。Metacat 表示"Metadata Catalog"，Metacat 伺服器允許生態學家利用網際網路分享他們的資料和後設資料。透過使用 Morpho 這個資料管理工具—生態學家便能容易的

上傳、下載、儲存、查詢和觀看儲存於 Metacat 伺服器裏的相關後設資料和資料。

Morpho 基本的操作功能包含有：產生和編輯後設資料、搜尋和查詢後設資料彙集、觀看資料與後設資料彙集、校對和編輯資料與後設資料、提供存取控制等。Morpho 所儲存的資料類型，包括了像資料表和影像檔案等。資料套件(data package)是 Morpho 產生能呈現資料檔案及其描述的邏輯單位(即文件(documentation)，或稱後設資料)。如前所述，Morpho 提供了一個簡單的操作介面，也就是經由這些資料套件來取得你所儲存的資料和所產生的文件。Morpho 的二個主要組成分別為資料套件精靈(Data Package Wizard)和 Morpho 編輯器(Editor)，資料套件精靈可藉由一步一步的指引操作產生一個新的資料套件；而 Morpho 編輯器則幫助我們在現有的資料套件上進行操作、編輯、和新增文件。如此將可協助研究人員簡單與快速的編輯 EML 文件與日後的存取修改等工作。

透過資料套件精靈將幫助我們產生符合資料套件所需最低標準的文件，這個過程包括 15 個簡單的步驟，每個步驟都利用不同的視窗來分開。套件精靈在必須要填寫的欄位利用紅色標識出來，但是記得若你提供的文件(即後設資料)越多越詳盡，更加容易使它成為科學家(也包括你自己)的瞭解和運用。紅色標識的欄位如果沒有填寫，則無法進行下一步驟的編輯。不過，記住我們可以利用 Morpho 編輯器隨時變更這些必須填寫或者其他相關的欄位資訊。事實上，在每一個步驟上都有填寫的說明與建議事項，這也是 Morpho 為何容易使用，加上還有線上「Morpho Help」，也就是 Morpho 的使用手冊，其中有非常詳盡的內容介紹每一操作步驟與相關注意事項。



Morpho 作業視窗進入畫面

New Data Package Wizard

Title and Abstract

Enter the title of the data package. The title field provides a description of the data that is long enough to differentiate it from other similar data.
e.g. Vernal Pool Amphibian Density Data, Isla Vista, CA USA, 1990-1996

Title:

Enter an abstract that describes the data package. This abstract is a paragraph or more that describes the particular data that are being documented. You may want to describe the objectives, key aspects, design or methods of the study.

Abstract:

Step 2 of 15

Cancel < Back Next > Finish

Morpho 資料套件精靈輸入視窗

Data Package: KNB.11.1

File Edit Search Documentation Data Window Help

Life history of the threatened Karkloof blue butterfly *Orachrysops ariadne*
 Accession Number: KNB.11.1 Keywords: *Orachrysops ariadne*, KwaZulu-Natal, *Camponotus natalensis*, life history, *Indigofera woodii*, Karkloof blue

Status of eggs of *Orachrysops ariadne* (number and percentage) from March to May from 1997 to 1999 at The Start.

| Year | Hatched | Damaged | Disappeared | Unknown | Total |
|------|-----------|----------|-------------|----------|----------|
| 1997 | 94(42.7) | 53(24.1) | 41(18.6) | 32(14.6) | 220(100) |
| 1998 | 122(56.7) | 49(22.4) | 40(21.9) | -- | 219(100) |
| 1999 | 90(55.2) | 44(27.0) | 29(17.8) | -- | 163(100) |

Entity/Attribute

Entity Description

Identifier: KNB.11.1
 Catalog: knb
 System: **Status of eggs of *Orachrysops ariadne* (number and percentage) from March to May from 1997 to 1999 at The Start.**
 Name: Status of eggs of *Orachrysops ariadne* (number and percentage) from March to May from 1997 to 1999 at The Start.
 Description: Status of eggs of *Orachrysops ariadne* (number and percentage) from March to May from 1997 to 1999 at The Start.

Physical Structure Description

Object Name: egg1.csv
 Muxum: Record column
 Length: Simple Field
 Text Format: Simple Delimited

資料及後設資料建置後之 Morpho 視窗畫面

Search

Query Title: Untitled-Search-1 ☒ Catalog Search ☒ Local Search

Subject Taxonomic Spatial Options

Check boxes determine which metadata fields are searched.

☒ Title
☐ All ☐ Abstract contains butterfly
☐ Keywords

Check boxes determine which metadata fields are searched.

☒ Title
☒ All ☒ Abstract contains
☐ And ☒ Or More Fewer

☐ Combine constraints from all tabs

Search Cancel

Morpho 查詢視窗畫面

四、心得

對生態學家來說，到野外辛苦的蒐集資料並不是一件困難的事，但所蒐集的資料如何能在日後發揮更大的效益，而不是當研究完成後，發表一兩篇報告後資料的價值便降低甚至消失。EML 的規劃與實施，將是生態資料分享與整合最基礎的資訊建設工作之一，可使得以往生態研究之資料處理、分享與整合的困難逐漸獲得改善，也打破以往個人蒐集資料、儲存及分析使用後資料流失的困境，使得資料之再利用及創造更珍貴的價值得以實現。

有效的資料庫整合方式是在研究過程的一開始，各個研究人員當提供他們的資料做為資料倉儲或歸檔用時便能符合 EML 的規格與相關技術。很不幸地，生態學家對資料庫的用途與使用卻存在許多的障礙有待克服。此外，這樣的整合技術將可能造成生態學家在研究方式上產生許多改變，而造成實行上的困難或配合意願低落。因此要發展出適當的技術，將必須對生態學研究過程有詳細的了解，同時也需要資訊人員、資訊管理員和生態學家三方之間密切的合作才得以完成。

雖然我們也逐漸體認到，資料長期的價值將取決於我們完善的後設資料記錄與管理方式，但這種將符合 EML 規格的資料歸檔工作，卻是卻是非常耗費時間與精力的工作，也將是我們面臨資訊整合的最大瓶頸。雖然生態資料的複雜性、異質性使得資料整合困難重重，但面臨資料分享與整合所產生的好處與效益，將促使研究人員樂於這種轉變才對。加上已有容易操作的介面 Morpho 發展出來，協助我們建立符合 EML 規格的資料與後設資料，今日不做，更待何時呢！

五、建議事項

- (一)、生態資料共享機制的建立：應有更多的經費與人力投入生態資訊管理的領域，加強生態資訊管理與共享機制的研究。研究人員對資料共享機制所涉及之各層面問題應仔細考量與充分討論後形成共識，例如對資料提供者的協定與權益保障、資料使用者之規範等問題。如此才可能讓生態資訊管理與共享機制得以落實，達成資料共享之優質研究機制。
- (二)、EML 的實施宜採漸進達成的方式：資料的分享因涉及所有權問題，目前研究人員對資料提供仍有許多疑慮，但對於應用 Morpho 在自己的電腦上來建置與管理個人或各機關之資料則有較高的意願，也是可行的第一步。如此先確保個人資料建置符合 EML 的規範，也就是至少先完成第一、二級資料鑑別與發現階段的標準，日後逐步改善 EML 的後續實施而達到完全符合 EML 文件化的標準，而達到最終資料整合的目標
- (三)、Morpho 的使用與推廣：借用美國發展出的這套專為生態學家設計的資料管理工具，Morpho 是一個容易使用與跨平台應用的介面，非常適合生態研究人員建置與管理個人的資料，可廣為推廣其應用。雖然並未發展出中文使用介面、無法推廣至一般社會大眾皆可使用，但這套軟體發展的主要目的是為了能促進資料的整合與共享，未來要能與國際交流與交換資料，還是應以英文建置為主。

Ecological Metadata Language (EML) Specification

Table of Contents

1. Preface

- 1.1. Introduction
- 1.2. Purpose Statement
- 1.3. Features

2. Overview of EML modules and their use

- 2.1. Module Overview Foreword
- 2.2. Root-level structure
 - 2.2.1. The eml module - A metadata container
 - 2.2.2. The eml-resource module - Base information for all resources
- 2.3. Top-level resources
 - 2.3.1. The eml-dataset module - Dataset specific information
 - 2.3.2. The eml-literature module - Citation specific information
 - 2.3.3. The eml-software module - Software specific information
 - 2.3.4. The eml-protocol module - Research protocol specific information
- 2.4. Supporting Modules - Adding detail to top-level resources
 - 2.4.1. The eml-access module - Access control rules for resources
 - 2.4.2. The eml-physical module - Physical file format
 - 2.4.3. The eml-party module - People and organization information
 - 2.4.4. The eml-coverage module - Geographic, temporal, and taxonomic extents of resources
 - 2.4.5. The eml-project module - Research context information for resources
 - 2.4.6. The eml-methods module - Methodological information for resources
- 2.5. Data organization - Modules describing dataset structures
 - 2.5.1. The eml-entity module - Entity level information within datasets
 - 2.5.2. The eml-attribute module - Attribute level information within dataset entities
 - 2.5.3. The eml-constraint module - Relationships among and within dataset entities
- 2.6. Entity types - Detailed information for discipline specific entities
 - 2.6.1. The eml-dataTable module - Logical information about data table entities
 - 2.6.2. The eml-spatialRaster module - Logical information about regularly gridded geospatial image data
 - 2.6.3. The eml-spatialVector module - Logical information about non-

gridded geospatial image data

2.6.4. Schema for validating spatial referencing descriptions.

2.6.5. The eml-storedProcedure module - Data tables resulting from procedures stored in a database

2.6.6. The eml-view module - Data tables resulting from a database query

2.7. Utility modules - Metadata documentation enhancements

2.7.1. The eml-text module - Text field formatting

2.7.2. Dependency Chart

3. Technical Architecture (Normative)

3.1. Introduction

3.2. Module Structure

3.3. Reusable Content

3.3.1. ID and Scope Examples

4. Module Descriptions (Normative)

4.1. eml

4.2. eml-access

4.3. eml-attribute

4.4. eml-constraint

4.5. eml-coverage

4.6. eml-dataset

4.7. eml-dataTable

4.8. eml-entity

4.9. eml-literature

4.10. eml-methods

4.11. eml-party

4.12. eml-physical

4.13. eml-project

4.14. eml-protocol

4.15. eml-resource

4.16. eml-software

4.17. eml-spatialRaster

4.18. eml-spatialReference

4.19. eml-spatialVector

4.20. eml-storedProcedure

4.21. eml-text

4.22. eml-view

Index

List of Examples

3.1. Invalid EML due to duplicate identifiers

3.2. Invalid EML due to a non-existent reference

3.3. Invalid EML due to a conflicting id attribute and a <references> element

3.4. A valid EML document

Chapter 1. Preface

Table of Contents

- 1.1. **Introduction**
- 1.2. **Purpose Statement**
- 1.3. **Features**

1.1. Introduction

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). EML is implemented as a series of XML document types that can be 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.

1.2. Purpose Statement

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.

1.3. Features

The architecture of EML was designed to serve the needs of the ecological community, and has benefitted 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.

The following list represents some of the features of EML:

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, a driving question throughout the 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 its 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 Consortium'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 meet 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.

There is a distinction between 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.

Chapter 2. Overview of EML modules and their use

Table of Contents

2.1. **Module Overview Foreword**

- 2.2. **Root-level structure**
 - 2.2.1. **The eml module - A metadata container**
 - 2.2.2. **The eml-resource module - Base information for all resources**
- 2.3. **Top-level resources**
 - 2.3.1. **The eml-dataset module - Dataset specific information**
 - 2.3.2. **The eml-literature module - Citation specific information**
 - 2.3.3. **The eml-software module - Software specific information**
 - 2.3.4. **The eml-protocol module - Research protocol specific information**
- 2.4. **Supporting Modules - Adding detail to top-level resources**
 - 2.4.1. **The eml-access module - Access control rules for resources**
 - 2.4.2. **The eml-physical module - Physical file format**
 - 2.4.3. **The eml-party module - People and organization information**
 - 2.4.4. **The eml-coverage module - Geographic, temporal, and taxonomic extents of resources**
 - 2.4.5. **The eml-project module - Research context information for resources**
 - 2.4.6. **The eml-methods module - Methodological information for resources**
- 2.5. **Data organization - Modules describing dataset structures**
 - 2.5.1. **The eml-entity module - Entity level information within datasets**
 - 2.5.2. **The eml-attribute module - Attribute level information within dataset entities**
 - 2.5.3. **The eml-constraint module - Relationships among and within dataset entities**
- 2.6. **Entity types - Detailed information for discipline specific entities**
 - 2.6.1. **The eml-dataTable module - Logical information about data table entities**
 - 2.6.2. **The eml-spatialRaster module - Logical information about regularly gridded geospatial image data**
 - 2.6.3. **The eml-spatialVector module - Logical information about non-gridded geospatial image data**
 - 2.6.4. **Schema for validating spatial referencing descriptions.**
 - 2.6.5. **The eml-storedProcedure module - Data tables resulting from procedures stored in a database**
 - 2.6.6. **The eml-view module - Data tables resulting from a database query**
- 2.7. **Utility modules - Metadata documentation enhancements**
 - 2.7.1. **The eml-text module - Text field formatting**
 - 2.7.2. **Dependency Chart**

2.1. Module Overview Foreword

The following section briefly describes each EML module and how they are logically designed in order to document ecological resources. Some of the modules are dependent on others, while others may be used as stand-alone descriptions. This section describes the modules using a "top down" approach, starting from the top-level eml wrapper module, followed by modules of

increasing detail. However, there are modules that may be used at many levels, such as eml-access. These modules are described when it is appropriate.

2.2. Root-level structure

2.2.1. The eml module - A metadata container

The eml module is a wrapper container that allows the inclusion of any metadata content in a single EML document. The eml module is used as a container to hold structured descriptions of ecological resources. In EML, the definition of a resource comes from the *The Dublin Core Metadata Initiative*, which describes a general element set used to describe "networked digital resources". The top-level structure of EML has been designed to be compatible with the Dublin Core syntax. In general, dataset resources, literature resources, software resources, and protocol resources comprise the list of information that may be described in EML. EML is largely designed to describe digital resources, however, it may also be used to describe non-digital resources such as paper maps and other non-digital media. *In EML, the definition of a "Data Package" is the combination of both the data and metadata for a resource.* So, data packages are built by using the <eml> wrapper, which will include all of the metadata, and optionally the data (or references to them). All EML packages must begin with the <eml> tag and end with the </eml> tag.

The eml module may be extended to describe other resources by means of its optional sub-field, <additionalMetadata>. This field is largely reserved for the inclusion of metadata that may be highly discipline specific and not covered in this version of EML, or it may be used to internally extend fields within the EML standard.

2.2.2. The eml-resource module - Base information for all resources

The eml-resource module contains general information that describes dataset resources, literature resources, protocol resources, and software resources. Each of the above four types of resources share a common set of information, but also have information that is unique to that particular resource type. Each resource type uses the eml-resource module to document the information common to all resources, but then extend eml-resource with modules that are specific to that particular resource type. For instance, all resources have creators, titles, and perhaps keywords, but only the dataset resource would have a "data table" within it. Likewise, a literature resource may have an "ISBN" number associated with it, whereas the other resource types would not.

The eml-resource module is exclusively used by other modules, and is therefore not a stand-alone module.

2.3. Top-level resources

The following four modules are used to describe separate resources: datasets, literature, software, and protocols. However, note that the dataset module makes use of the other top-level modules by importing them at different levels. For instance, a dataset may have been produced using a particular protocol, and that protocol may come from a protocol document in a library of protocols. Likewise, citations are used throughout the top-level resource modules by importing the literature module.

2.3.1. The eml-dataset module - Dataset specific information

The eml-dataset module contains general information that describes dataset resources. It is intended to provide overview information about the dataset: broad information such as the title, abstract, keywords, contacts, maintenance history, purpose, and distribution of the data themselves. The eml-dataset module also imports many other modules that are used to describe the dataset in fine detail. Specifically, it uses the eml-methods module to describe methodology used in collecting or processing the dataset, the eml-project module to describe the overarching research context and experimental design, the eml-access module to define access control rules for the data and metadata, and the eml-entity module to provide detailed information about the logical structure of the dataset. A dataset can be (and often is) composed of a series of data entities (tables) that are linked together by particular integrity constraints.

The eml-dataset module, like other modules, may be "referenced" via the <references> tag. This allows a dataset to be described once, and then used as a reference in other locations within the EML document via its ID.

2.3.2. The eml-literature module - Citation specific information

The eml-literature module contains information that describes literature resources. It is intended to provide overview information about the literature citation, including title, abstract, keywords, and contacts. Citation types follow the conventions laid out by **EndNote**, and there is an attempt to represent a compatible subset of the EndNote citation types. These citation types include: article, book, chapter, edited book, manuscript, report, thesis, conference proceedings, personal communication, map, generic, audio visual, and presentation. The "generic" citation type would be used when one of the other types will not work.

The eml-literature module, like other modules, may be "referenced" via the <references> tag. This allows a citation to be described once, and then used as a reference in other locations within the EML document via its ID.

2.3.3. The eml-software module - Software specific information

The eml-software module contains general information that describes software resources. This module is intended to fully document software that is needed in order to view a resource (such as a dataset) or to process a dataset. The software module is also imported into the eml-methods module in order to document what software was used to process or perform quality control procedures on a dataset.

The eml-software module, like other modules, may be "referenced" via the <references> tag. This allows a software resource to be described once, and then used as a reference in other locations within the EML document via its ID.

2.3.4. The eml-protocol module - Research protocol specific information

The EML Protocol Module is used to define abstract, prescriptive procedures for generating or processing data. Conceptually, a protocol is a standardized method.

Eml-protocol resembles eml-methods; however, eml-methods is descriptive (often written in the declarative mood: "I took five subsamples...") whereas eml-protocol is prescriptive (often written in the imperative mood: "Take five subsamples..."). A protocol may have versions, whereas methods (as used in eml-methods) should not.

2.4. Supporting Modules - Adding detail to top-level resources

The following six modules are used to qualify the resources being described in more detail. They are used to describe access control rules, distribution of the metadata and data themselves, parties associated with the resource, the geographic, temporal, and taxonomic extents of the resource, the overall research context of the resource, and detailed methodology used for creating the resource. Some of these modules are imported directly into the top-level resource modules, often in many locations in order to limit the scope of the description. For instance, the eml-coverage module may be used for a particular column of a dataset, rather than the entire dataset as a whole.

2.4.1. The eml-access module - Access control rules for resources

The eml-access module describes the level of access that is to be granted or denied to a resource (including both metadata and data) for a particular user or group of users. The eml-access module uses references to a particular authentication system to determine the set of principals (users + groups) that can be specified in access rules, and the membership of users in groups. The special principal 'public' can be used to indicate that any user or group has a particular access permission, thereby making it easier to specify that anonymous access is allowed.

The rules specified in this module will determine the level of access to a resource for specific users and groups. There are two mechanisms for including access control information via the eml-access module: 1) Each top-level resource module (eml-dataset, eml-literature, eml-software, and eml-protocol) includes an optional <access> element that is used to define the default access control at the resource level for all metadata and data in the EML package. If this access element is omitted from the document, then by definition the package submitter should be given full access to the package but all other users should be denied all access. 2) Exceptions for particular data components of the package can be controlled at a finer grain by using an access description in the <additionalMetadata> element that points at a particular <distribution> element by referencing its unique id. An access control document may be defined, or referenced, from this location, and the <describes> element is used to point to the distribution that is to be controlled via its "id" attribute. When access control rules are specified this way, they apply to the data in the distribution element, and not the metadata. Thus, it controls access to the content of the <inline> element, as well as resources that are referenced by the <online/url> and <online/connection> elements. These exceptions to access to particular data resources are applied after the default access rules from (1) are applied, so they effectively override the default rules when they overlap. Any access rules that are specified in locations other than the top-level 'access' element or in 'additionalMetadata' pointing at only a distribution id are completely ignored. For example, rules found in '/eml/dataset/methods/methodStep/citation/access' would be completely ignored.

For describing access to different data entities, one can either provide multiple <describes> elements within a single additionalMetadata element, or provide multiple additionalMetadata elements, each of which describes access exceptions for a particular distribution. Note that each 'additionalMetadata' element can only contain one child element besides the describes tags (e.g., only one <access> element).

The values that can be assigned to the <permission> element are: "read", "write", "changePermission", and "all". Users given read permission can view the resource; write can make changes to the resource excluding changes to access rules; changePermission can change access rules; and all has all of read, write, and changePermission.

An example is given below, with non-critical sections elided:

```
<eml>
  <dataset>
    <access>
      authSystem="ldap://ldap.ecoinformatics.org:389/dc=ecoinformatics,dc=org"
      order="allowFirst">
    <allow>
      <principal>uid=alice,o=NASA,dc=ecoinformatics,dc=org</principal>
      <permission>read</permission>
      <permission>write</permission>
    <allow>
  </access>
  ...
</dataset>
<additionalMetadata>
  <describes>entity123</describes>
  <describes>entity234</describes>
  <access>
    authSystem="ldap://ldap.ecoinformatics.org:389/dc=ecoinformatics,dc=org"
```

```

    order="allowFirst">
<deny>
  <principal>uid=alice,o=NASA,dc=ecoinformatics,dc=org</principal>
  <permission>write</permission>
</deny>
</access>
</additionalMetadata>
<eml>

```

In this example, the overall default access is for the user alice to be able to read and write all metadata and data, but nobody else has any access beside the submitter. However, for entity123 and entity234, there is an additional rule saying that user alice does not have write permission. The net effect is that alice can read and make changes to the metadata, but can only read the two data entities.

The eml-access module, like other modules, may be "referenced" via the <references> tag. This allows an access control document to be described once, and then used as a reference in other locations within the EML document via its ID.

In summary, access rules can be applied in two places in an eml document, and all other 'access' elements in the document are completely ignored. Default access rules are established in the top 'access' element for the main eml resource (e.g., "/eml/dataset/access"). These default rules can be overridden for particular data entities by providing an additional "access" element in <additionalMetadata> with a <describes> element that points at the id of the distribution for the data entity.

2.4.2. The eml-physical module - Physical file format

The eml-physical module describes the external and internal physical characteristics of a data object as well as the information required for its distribution. Examples of the external physical characteristics of a data object would be the filename, size, compression, encoding methods, and authentication of a file or byte stream. Internal physical characteristics describe the format of the data object being described. Both named binary or otherwise proprietary formats can be cited (e.g., Microsoft Access 2000), or text formats can be precisely described (e.g., ASCII text delimited with commas). For these text formats, it also includes the information needed to parse the data object to extract the entity and its attributes from the data object. Distribution information describes how to retrieve the data object. The retrieval information can be either online (e.g., a URL or other connection information) or offline (e.g., a data object residing on an archival tape).

The eml-physical module, like other modules, may be "referenced" via the <references> tag. This allows a physical document to be described once, and then used as a reference in other locations within the EML document via its ID.

2.4.3. The eml-party module - People and organization information

The eml-party module describes a responsible party (person or organization), and is typically used to name the originator of a resource or metadata document. It contains detailed contact information for the party, be it an individual person, an organization, or a named position within an organization. The eml-party module is used throughout the other EML modules where detailed contact information is needed.

The eml-party module, like other modules, may be "referenced" via the <references> tag. This allows a party to be described once, and then used as a reference in other locations within the EML document via its ID.

2.4.4. The eml-coverage module - Geographic, temporal, and taxonomic extents of resources

The eml-coverage module contains fields for describing the coverage of a resource in terms of time, space, and taxonomy. These coverages (temporal, spatial, and taxonomic) represent the extent of applicability of the resource in those domains. The Geographic coverage section allows for 2 means of expressing coverage on the surface of the earth: 1) via a set of bounding coordinates that define the North, South, East and West points in a rectangular area, optionally including a bounding altitude, and 2) using a G-Ring polygon definition, where an irregularly shaped area may be defined using an ordered list of latitude/longitude coordinates. A G-Ring may also include an "inner G-Ring" that defines one or more "cut-outs" in the area, i.e. the donut hole concept.

The temporal coverage section allows for the definition of either a single date/time, or a range of dates/times. These date/times may be expressed as a calendar date according to the ISO 8601 Date and Time Specification, or by using an alternate time scale, such as the geologic time scale. In order to express an "ongoing" time frame, the end date in the range would likely use the alternate time scale fields with a value of "ongoing", whereas the begin date would use the specific calendar date fields.

The taxonomic coverage section allows for detailed description of the taxonomic extent of the dataset or resource. The taxonomic classification consists of a recursive set of taxon rank names, their values, and their common names. This construct allows for a taxonomic hierarchy to be built to show the level of identification (e.g. Rank Name = Kingdom, Rank Value = Animalia, Common Name = Animals, and so on down the hierarchy.) The taxonomic coverage module also allows for the definition of the classification system in cases where alternative systems are used.

The eml-coverage module, like other modules, may be "referenced" via the <references> tag. This allows the coverage extent to be described once, and then used as a reference in other locations within the EML document via its ID.

2.4.5. The eml-project module - Research context information for resources

The eml-project module describes the research context in which the dataset was created, including descriptions of over-all motivations and goals, funding, personnel, description of the

study area etc. This is also the module to describe the design of the project: the scientific questions being asked, the architecture of the design, etc. This module is used to place the dataset that is being documented into its larger research context.

The eml-project module, like other modules, may be "referenced" via the <references> tag. This allows a research project to be described once, and then used as a reference in other locations within the EML document via its ID.

2.4.6. The eml-methods module - Methodological information for resources

The eml-methods module describes the methods followed in the creation of the dataset, including description of field, laboratory and processing steps, sampling methods and units, quality control procedures. The eml-methods module is used to describe the *actual* procedures that are used in the creation or the subsequent processing of a dataset. Likewise, eml-methods is used to describe processes that have been used to define / improve the quality of a data file, or to identify potential problems with the data file. Note that the eml-protocol module is intended to be used to document a *prescribed* procedure, whereas the eml-method module is used to describe procedures that *were actually performed*. The distinction is that the use of the term "protocol" is used in the "prescriptive" sense, and the term "method" is used in the "descriptive" sense. This distinction allows managers to build a protocol library of well-known, established protocols (procedures), but also document what procedure was truly performed in relation to the established protocol. The method may have diverged from the protocol purposefully, or perhaps incidentally, but the procedural lineage is still preserved and understandable.

2.5. Data organization - Modules describing dataset structures

The following three modules are used to document the logical layout of a dataset. Many datasets are comprised of multiple entities (e.g. a series of tabular data files, or a set of GIS features, or a number of tables in a relational database). Each entity within a dataset may contain one or more attributes (e.g. multiple columns in a datafile, multiple attributes of a GIS feature, or multiple columns of a database table). Lastly, there may be both simple or complex relationships among the entities within a dataset. The relationships, or the constraints that are to be enforced in the dataset, are described using the eml-constraint module. All entities share a common set of information (described using eml-entity), but some discipline specific entities have characteristics that are unique to that entity type. Therefore, the eml-entity module is extended for each of these types (dataTable, spatialRaster, spatialVector, etc...) which are described in the next section.

2.5.1. The eml-entity module - Entity level information within datasets

The eml-entity module defines the logical characteristics of each entity in the dataset. Entities are usually tables of data (eml-dataTable). Data tables may be ascii text files, relational database tables, spreadsheets or other type of tabular data with a fixed logical structure. Related to data tables are views (eml-view) and stored procedures (eml-storedProcedure). Views and stored procedures are produced by an RDBMS or related system. Other types of data such as: raster (eml-spatialRaster), vector (eml-spatialVector) or spatialReference image data are also data entities. An otherEntity element would be used to describe types of entities that are not described by any other entity type.

The eml-entity module, like other modules, may be "referenced" via the <references> tag. This allows an entity document to be described once, and then used as a reference in other locations within the EML document via its ID.

2.5.2. The eml-attribute module - Attribute level information within dataset entities

The eml-attribute module describes all attributes (variables) in a data entity: dataTable, spatialRaster, spatialVector, storedProcedure, view or otherEntity. The description includes the name and definition of each attribute, its domain, definitions of coded values, and other pertinent information. Two structures exist in this module: 1. attribute is used to define a single attribute; 2. attributeList is used to define a list of attributes that go together in some logical way.

The eml-attribute module, like other modules, may be "referenced" via the <references> tag. This allows an attribute document to be described once, and then used as a reference in other locations within the EML document via its ID.

Philosophy of Attribute Units

The concept of "unit" represents one of the most fundamental categories of metadata. The classic example of data entropy is the case in which a reported numeric value loses meaning due to lack of associated units. Much of Ecology is driven by measurement, and most measurements are inherently comparative. Good data description requires a representation of the basis for comparison, i.e., the unit. In modeling the attribute element, the authors of EML drew inspiration from the **NIST Reference on Constants, Units, and Uncertainty**. This document defines a unit 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." The authors of the EML 2.0 specification (hereafter "the authors") decided to make the unit element required, wherever possible.

Units may also be one of the most problematic categories of metadata. For instance, there are many candidate attributes that clearly have no units, such as named places and letter grades. There are other candidate attributes for which units are difficult to identify, despite some suspicion that they should exist (e.g. pH, dates, times). In still other cases, units 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.

The authors decided to sharpen the model of attribute by nesting unit under measurementScale. Measurement Scale is a data typology, borrowed from Statistics, that was introduced in the 1940's. Under the adopted model, attributes are classified as nominal,

ordinal, interval, and ratio. Though widely criticized, this classification is well-known and provides at least first-order utility in EML. For example, nesting unit under measurementScale allows EML to prevent its meaningless inclusion for categorical data -- an approach judged superior to making unit universally required or universally optional.

The sharpening of the attribute model allowed the elimination of the unit type "undefined" from the standard unit dictionary (see eml-unitDictionary.xml). It seemed self-defeating to require the unit element exactly where appropriate, yet still allow its content to be undefined. An attribute that requires a unit definition is malformed until one is provided. The unit type "dimensionless" is preserved, however. In EML 2.0, it 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 adopted measurement scale typology.

Closely related to the concept of unit is the concept of attribute domain. The authors decided that a well-formed description of an attribute must include some indication of the set of possible values for that attribute. The set of possible values is useful, perhaps necessary, for interpreting any particular observed value. While universally required, attribute domain has different forms, depending on the associated measurement scale.

The element storageType has an obvious relationship to domain. It gives some indication of the range of possible values of an attribute, and also gives some (potentially critical) operability information about the way the attribute is represented or construed in the local storage system. The storageType element seems to fall in a gray area between the logical and physical aspects of stored data. Neither comfortable with eliminating it nor with making it required, the authors left it available but optional under attribute. In addition, it is repeatable so that different storage types can be provided for various systems (e.g., different databases might use different types for columns, even though the domain of the attribute is the same regardless of which database is used).

Attributes representing dates, times, or combinations thereof (hereafter "dateTime") were the most difficult to model in EML. Is dateTime of type interval or ordinal? Does it have units or not? Strong cases can be made on each side of the issue. The confusion may reflect the limitations of the measurement scale typology. The final resolution of the dateTime model is probably somewhat arbitrary. There was clearly a need, however, to allow for the interoperability of dateTime formats. EML 2.0 tries to provide an unambiguous mechanism for describing the format of dateTime values by providing a separate category for datetime values. This "datetime" measurement scale allows users to explicitly label attributes that contain Gregorian date and time values, and allows them to provide the information needed to parse these values into their appropriate components (e.g., days, months, years)./

2.5.3. The eml-constraint module - Relationships among and within dataset entities

The eml-constraint schema defines the integrity constraints between entities (e.g., data tables) as they would be maintained in a relational management system. These constraints include primary key constraints, foreign key constraints, unique key constraints, check constraints, and not null constraints, among potential others.

2.6. Entity types - Detailed information for discipline specific entities

The following six modules are used to describe a number of common types of entities found in datasets. Each entity type uses the eml-entity module elements as it's base set of elements, but then extends the base with entity-specific elements. Note that the eml-spatialReference module is not an entity type, but is rather a common set of elements used to describe spatial reference systems in both eml-spatialRaster and eml-spatialVector. It is described here in relation to those two modules.

2.6.1. The eml-dataTable module - Logical information about data table entities

The eml-dataTable module is used to describe the logical characteristics of each tabular set of information in a dataset. A series of comma-separated text files may be considered a dataset, and each file would subsequently be considered a dataTable entity within the dataset. Since the eml-dataTable module extends the eml-entity module, it uses all of the common entity elements to describe the table, along with a few elements specific to just data table entities. The eml-dataTable module allows for the description of each attribute (column/field/variable) within the data table through the use of the eml-attribute module. Likewise, there are fields used to describe the physical distribution of the data table, its overall coverage, the methodology used in creating the data, and other logical structure information such as its orientation, case sensitivity, etc.

2.6.2. The eml-spatialRaster module - Logical information about regularly gridded geospatial image data

The eml-spatialRaster module allows for the description of entities composed of rectangular grids of data values that are usually georeferenced to a portion of the earth's surface. Specific attributes of a spatial raster can be documented here including the spatial organization of the raster cells, the cell data values, and if derived via imaging sensors, characteristics about the image and its individual bands.

2.6.3. The eml-spatialVector module - Logical information about non-gridded geospatial image data

The eml-spatialVector module allows for the description of spatial objects in a GIS system that are not defined in a regularly gridded pattern. These geometries include points and vectors and the relationships among them. Specific attributes of a spatial vector can be documented here including the vector's geometry type, count and topology level.

2.6.4. Schema for validating spatial referencing descriptions.

This module defines both projected and uprojected coordinate systems for referencing the spatial coordinates of a dataset to the earth. The schema is based on that used by Environmental Systems Research Inc (ESRI) for its .prj file format. EML provides a library of pre-defined coordinate systems that may be referred to by name in the horizCoordSysName element. A custom projection may be defined using this schema for any projection that does not appear in this dictionary.

2.6.5. The eml-storedProcedure module - Data tables resulting from procedures stored in a database

The StoredProcedure module is meant to capture information on procedures that produce data output in the form of a data table. In an RDBMS one can code complex queries and transactions into stored procedures and then invoke them directly from front-end applications. It allows the optional description of any parameters that are expected to be passed to the procedure when it is called.

2.6.6. The eml-view module - Data tables resulting from a database query

The eml-view module describes a view from a database management system. A view is a query statement that is stored as a database object and executed each time the view is called.

2.7. Utility modules - Metadata documentation enhancements

The following modules are used to highlight the information being documented in each of the above modules where prose may be needed to convey the critical metadata. The eml-text module provides a number of text-based constructs to enhance a document (including sections, paragraphs, lists, subscript, superscript, emphasis, etc.)

2.7.1. The eml-text module - Text field formatting

The eml-text module is a wrapper container that allows general text descriptions to be used within the various modules of eml. It can include either structured or unstructured text blocks. It isn't really appropriate to use this module outside of the context of a parent module, because the parent module determines the appropriate context to which this text description applies. The eml-text module allows one to provide structure to a text description in order to convey concepts such as sections (paragraphs), hierarchy (ordered and unordered lists), emphasis (bold, superscript, subscript) etc. The structured elements are a subset of **DocBook** so the predefined DocBook stylesheets can be used to style EML fields that implement this module.

2.7.2. Dependency Chart

The multiple modules in EML all depend on each other in complex ways. To easily see these dependencies see the **EML Dependency Chart**.

Chapter 3. Technical Architecture (Normative)

Table of Contents

- 3.1. **Introduction**
- 3.2. **Module Structure**
- 3.3. **Reusable Content**
 - 3.3.1. **ID and Scope Examples**

3.1. Introduction

This section explains the rules of EML. There are some rules that cannot be written directly into the XML Schemas nor enforced by an XML parser. These are guidelines that every EML package must follow in order for it to be considered EML compliant.

3.2. Module Structure

Each EML module, with the exception of "eml" itself, has a top level choice between the structured content of that modules or a "references" field. This enables the reuse of content previously defined elsewhere in the document. Methods for defining and referencing content

are described in the **next** section

3.3. Reusable Content

EML allows the reuse of previously defined structured content (DOM sub-trees) through the use of key/keyRef type references. In order for an EML package to remain cohesive and to allow for the cross platform compatability of packages, the following rules with respect to packaging must be followed.

An ID is required on the eml root element.

IDs are optional on all other elements.

If an ID is not provided, that content must be interpreted as representing a distinct object.

If an ID is provided for content then that content is distinct from all other content except for that content that references its ID.

If a user wants to reuse content to indicate the repetition of an object, a reference must be used. Two identical ids cannot exist in a single document.

"Document" scope is defined as identifiers unique only to a single instance document (if a document does not have a system attribute or if scope is set to 'document' then all IDs are defined as distinct content).

"System" scope is defined as identifiers unique to an entire data management system (if two documents share a system string, then any IDs in those two documents that are identical refer to the same object).

If an element references another element, it must not have an ID itself.

All EML packages must have the 'eml' module as the root.

The system and scope attribute are always optional except for at the 'eml' module where the scope attribute is fixed as 'system'. The scope attribute defaults to 'document' for all other modules.

3.3.1. ID and Scope Examples

3.3.1.1. EML Parser

Because some of these rules cannot be enforced in XML-Schema, we have written a parser which checks the validity of the references and IDs used in your document. This parser is included with the 2.0.1 release of EML. To run the parser, you must have Java 1.3.1 or higher. To execute it change into the lib directory of the release and run the 'runEMLParser' script passing your EML instance file as a parameter. There is also an **online version** of this parser which is publicly accessible. The online parser will both validate your XML document against the schema as well as check the integrity of your references.

3.3.1.2. Example Documents

Example 3.1. Invalid EML due to duplicate identifiers

```
<?xml version="1.0"?>
<eml:eml
  packageId="eml.1.1" system="knbn"
  xmlns:eml="eml://ecoinformatics.org/eml-2.0.1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="eml://ecoinformatics.org/eml-2.0.1 eml.xsd">

  <dataset id="ds.1">
    <title>Sample Dataset Description</title>
    <!-- the two creators have the same id.  this should be an error-->
    <creator id="23445" scope="document">
      <individualName>
        <surName>Smith</surName>
      </individualName>
    </creator>
    <creator id="23445" scope="document">
      <individualName>
        <surName>Myer</surName>
      </individualName>
    </creator>
    ...
  </dataset>
</eml:eml>
```

This instance document is invalid because both creator elements have the same id. No two elements can have the same string as an id.

Example 3.2. Invalid EML due to a non-existent reference

```
<?xml version="1.0"?>
<eml:eml
  packageId="eml.1.1" system="knbn"
  xmlns:eml="eml://ecoinformatics.org/eml-2.0.1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="eml://ecoinformatics.org/eml-2.0.1 eml.xsd">

  <dataset id="ds.1">
    <title>Sample Dataset Description</title>
```

```

<creator id="23445" scope="document">
  <individualName>
    <surName>Smith</surName>
  </individualName>
</creator>
<creator id="23446" scope="document">
  <individualName>
    <surName>Myer</surName>
  </individualName>
</creator>
...
<contact>
  <references>23447</references>
</contact>
</dataset>
</eml:eml>

```

This instance document is invalid because the contact element references an id that does not exist. Any referenced id must exist.

Example 3.3. Invalid EML due to a conflicting id attribute and a <references> element

```

<?xml version="1.0"?>
<eml:eml
  packageId="eml.1.1" system="knbn"
  xmlns:eml="eml://ecoinformatics.org/eml-2.0.1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="eml://ecoinformatics.org/eml-2.0.1 eml.xsd">

  <dataset id="ds.1">
    <title>Sample Dataset Description</title>
    <creator id="23445" scope="document">
      <individualName>
        <surName>Smith</surName>
      </individualName>
    </creator>
    <creator id="23446" scope="document">
      <individualName>
        <surName>Meyer</surName>
      </individualName>
    </creator>
    ...
    <contact id="522">
      <references>23445</references>
    </contact>
  </dataset>
</eml:eml>

```

This instance document is invalid because the contact element both references another element and has an id itself. If an element references another element, it may not have an id. This prevents circular references.

Example 3.4. A valid EML document

```
<?xml version="1.0"?>
<eml:eml
  packageId="eml.1.1" system="knb"
  xmlns:eml="eml://ecoinformatics.org/eml-2.0.1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="eml://ecoinformatics.org/eml-2.0.1 eml.xsd">

  <dataset id="ds.1">
    <title>Sample Dataset Description</title>
    <creator id="23445" scope="document">
      <individualName>
        <surName>Smith</surName>
      </individualName>
    </creator>
    <creator id="23446" scope="document">
      <individualName>
        <surName>Smith</surName>
      </individualName>
    </creator>
    ...
    <contact>
      <references>23446</references>
    </contact>
    <contact>
      <references>23445</references>
    </contact>
  </dataset>
</eml:eml>
```

This instance document is valid. Each contact is referencing one of the creators above and all the ids are unique.

Chapter 4. Module Descriptions (Normative)

Table of Contents

4.1. **eml**

4.2. **eml-access**

- 4.3. **eml-attribute**
- 4.4. **eml-constraint**
- 4.5. **eml-coverage**
- 4.6. **eml-dataset**
- 4.7. **eml-dataTable**
- 4.8. **eml-entity**
- 4.9. **eml-literature**
- 4.10. **eml-methods**
- 4.11. **eml-party**
- 4.12. **eml-physical**
- 4.13. **eml-project**
- 4.14. **eml-protocol**
- 4.15. **eml-resource**
- 4.16. **eml-software**
- 4.17. **eml-spatialRaster**
- 4.18. **eml-spatialReference**
- 4.19. **eml-spatialVector**
- 4.20. **eml-storedProcedure**
- 4.21. **eml-text**
- 4.22. **eml-view**

4.1. eml

Normative technical docs for **eml**

4.2. eml-access

Normative technical docs for **eml-access**

4.3. eml-attribute

Normative technical docs for **eml-attribute**

4.4. eml-constraint

Normative technical docs for **eml-constraint**

4.5. eml-coverage

Normative technical docs for **eml-coverage**

4.6. eml-dataset

Normative technical docs for **eml-dataset**

4.7. eml-dataTable

Normative technical docs for **eml-dataTable**

4.8. eml-entity

Normative technical docs for **eml-entity**

4.9. eml-literature

Normative technical docs for **eml-literature**

4.10. eml-methods

Normative technical docs for **eml-methods**

4.11. eml-party

Normative technical docs for **eml-party**

4.12. eml-physical

Normative technical docs for **eml-physical**

4.13. eml-project

Normative technical docs for **[eml-project](#)**

4.14. **eml-protocol**

Normative technical docs for **[eml-protocol](#)**

4.15. **eml-resource**

Normative technical docs for **[eml-resource](#)**

4.16. **eml-software**

Normative technical docs for **[eml-software](#)**

4.17. **eml-spatialRaster**

Normative technical docs for **[eml-spatialRaster](#)**

4.18. **eml-spatialReference**

Normative technical docs for **[eml-spatialReference](#)**

4.19. **eml-spatialVector**

Normative technical docs for **[eml-spatialVector](#)**

4.20. **eml-storedProcedure**

Normative technical docs for **[eml-storedProcedure](#)**

4.21. **eml-text**

Normative technical docs for **eml-text**

4.22. eml-view

Normative technical docs for **eml-view**

Index

A

- abstract**-eml-project
- abstract**-eml-resource
- access**-eml-access
- access**-eml-dataset
- access**-eml-literature
- access**-eml-protocol
- access**-eml-software
- accuracy**-eml-attribute
- accuracyReport**-eml-spatialRaster
- accuracyReport**-eml-spatialVector
- action**-eml-software
- additionalInfo**-eml-entity
- additionalInfo**-eml-resource
- additionalMetadata**-eml
- address**-eml-party
- administrativeArea**-eml-party
- allow**-eml-access
- alternateIdentifier**-eml-entity
- alternateIdentifier**-eml-resource
- alternativeTimeScale**-eml-coverage
- altitudeDatumName**-eml-spatialReference
- altitudeDistanceUnits**-eml-spatialReference
- altitudeEncodingMethod**-eml-spatialReference
- altitudeMaximum**-eml-coverage
- altitudeMinimum**-eml-coverage
- altitudeResolution**-eml-spatialReference
- altitudeSysDef**-eml-spatialReference
- altitudeUnits**-eml-coverage
- article**-eml-literature
- associatedParty**-eml-resource
- attribute**-eml-attribute
- attribute**-eml-attribute

attributeAccuracyExplanation-eml-attribute
attributeAccuracyReport-eml-attribute
attributeAccuracyValue-eml-attribute
attributeDefinition-eml-attribute
attributeLabel-eml-attribute
attributeList-eml-attribute
attributeList-eml-dataTable
attributeList-eml-entity
attributeList-eml-spatialRaster
attributeList-eml-spatialVector
attributeList-eml-storedProcedure
attributeList-eml-view
attributeName-eml-attribute
attributeOrientation-eml-physical
attributeReference-eml-constraint
attributeReference-eml-constraint
attributeReference-eml-constraint
attributeReference-eml-constraint
attributeReference-eml-constraint
audioVisual-eml-literature
authentication-eml-physical

B

bandDescription-eml-spatialRaster
bandgapbytes-eml-physical
bandrowbytes-eml-physical
beginDate-eml-coverage
bilinearFit-eml-spatialRaster
binaryRasterFormat-eml-physical
book-eml-literature
bookTitle-eml-literature
boundingAltitudes-eml-coverage
boundingCoordinates-eml-coverage
bounds-eml-attribute
bounds-eml-attribute
byteorder-eml-physical

C

calendarDate-eml-coverage
cameraCalibrationInformationAvailability-eml-spatialRaster
cardinality-eml-constraint
caseSensitive-eml-dataTable

cellGeometry-eml-spatialRaster
cellSizeXDirection-eml-spatialRaster
cellSizeYDirection-eml-spatialRaster
changeDate-eml-dataset
changeHistory-eml-dataset
changeScope-eml-dataset
chapter-eml-literature
chapterNumber-eml-literature
characterEncoding-eml-physical
checkCondition-eml-constraint
checkConstraint-eml-constraint
checksum-eml-software
childOccurences-eml-constraint
citation-eml
citation-eml-attribute
citation-eml-literature
citation-eml-methods
citation-eml-methods
citation-eml-physical
citation-eml-project
citation-eml-project
citation-eml-project
city-eml-party
classificationSystem-eml-coverage
classificationSystemCitation-eml-coverage
classificationSystemModifications-eml-coverage
cloudCoverPercentage-eml-spatialRaster
code-eml-attribute
code-eml-attribute
codeDefinition-eml-attribute
codeExplanation-eml-attribute
codesetName-eml-attribute
codesetURL-eml-attribute
collapseDelimiters-eml-physical
collapseDelimiters-eml-physical
column-eml-spatialRaster
columns-eml-spatialRaster
comment-eml-dataset
commonName-eml-coverage
communicationType-eml-literature
complex-eml-physical
compressionGenerationQuality-eml-spatialRaster
compressionMethod-eml-physical
conferenceDate-eml-literature

conferenceDate-eml-literature
conferenceLocation-eml-literature
conferenceLocation-eml-literature
conferenceName-eml-literature
conferenceName-eml-literature
conferenceProceedings-eml-literature
connection-eml-physical
connection-eml-resource
connectionDefinition-eml-physical
connectionDefinition-eml-resource
connectionDefinition-eml-resource
constraint-eml-dataTable
constraint-eml-entity
constraint-eml-spatialRaster
constraint-eml-spatialVector
constraint-eml-storedProcedure
constraint-eml-view
constraintDescription-eml-constraint
constraintName-eml-constraint
contact-eml-dataset
controlPoint-eml-spatialRaster
corner-eml-spatialRaster
cornerPoint-eml-spatialRaster
country-eml-party
coverage-eml-attribute
coverage-eml-entity
coverage-eml-methods
coverage-eml-methods
coverage-eml-project
coverage-eml-resource
creator-eml-resource
customUnit-eml-attribute

D

dataFormat-eml-physical
dataset-eml
dataset-eml-dataset
datasetGPolygon-eml-coverage
datasetGPolygonExclusionGRing-eml-coverage
datasetGPolygonOuterGRing-eml-coverage
dataSource-eml-methods
dataTable-eml-dataset
dataTable-eml-dataTable

datetime-eml-attribute
dateTimeDomain-eml-attribute
dateTimePrecision-eml-attribute
datum-eml-spatialReference
defaultValue-eml-resource
definition-eml-attribute
definition-eml-attribute
definition-eml-resource
definitionAttributeReference-eml-attribute
degree-eml-literature
deliveryPoint-eml-party
deny-eml-access
dependency-eml-software
depthDatumName-eml-spatialReference
depthDistanceUnits-eml-spatialReference
depthEncodingMethod-eml-spatialReference
depthResolution-eml-spatialReference
depthSysDef-eml-spatialReference
describes-eml
description-eml-dataset
description-eml-methods
description-eml-methods
description-eml-project
description-eml-resource
descriptor-eml-project
descriptorValue-eml-project
designDescription-eml-project
diskUsage-eml-software
distribution-eml-physical
distribution-eml-resource
distribution-eml-software
domainDescription-eml-storedProcedure

E

eastBoundingCoordinate-eml-coverage
editedBook-eml-literature
edition-eml-literature
edition-eml-literature
edition-eml-literature
editor-eml-literature
electronicMailAddress-eml-party
eml-eml
emphasis-eml-text

encodingMethod-eml-physical
endDate-eml-coverage
entityCodeList-eml-attribute
entityDescription-eml-entity
entityName-eml-entity
entityReference-eml-attribute
entityReference-eml-constraint
entityType-eml-entity
enumeratedDomain-eml-attribute
externalCodeSet-eml-attribute
externallyDefinedFormat-eml-physical

F

fieldDelimiter-eml-physical
fieldDelimiter-eml-physical
fieldStartColumn-eml-physical
fieldWidth-eml-physical
filmDistortionInformationAvailability-eml-spatialRaster
foreignKey-eml-constraint
formatName-eml-physical
formatString-eml-attribute
formatVersion-eml-physical
funding-eml-project

G

generalTaxonomicCoverage-eml-coverage
generic-eml-literature
geogCoordSys-eml-spatialReference
geogCoordSys-eml-spatialReference
geographicCoverage-eml-coverage
geographicCoverage-eml-literature
geographicDescription-eml-coverage
geometricObjectCount-eml-spatialVector
geometry-eml-spatialVector
georeferenceInfo-eml-spatialRaster
givenName-eml-party
gRing-eml-coverage
gRing-eml-coverage
gRingLatitude-eml-coverage
gRingLongitude-eml-coverage
gRingPoint-eml-coverage
gRingPoint-eml-coverage

H

highWavelength-eml-spatialRaster
horizCoordSysDef-eml-spatialReference
horizCoordSysDef-eml-spatialReference
horizCoordSysName-eml-spatialReference
horizontalAccuracy-eml-spatialRaster
horizontalAccuracy-eml-spatialVector

I

identificationReference-eml-coverage
identifierName-eml-coverage
illuminationAzimuthAngle-eml-spatialRaster
illuminationElevationAngle-eml-spatialRaster
imageDescription-eml-spatialRaster
imageOrientationAngle-eml-spatialRaster
imageQualityCode-eml-spatialRaster
imagingCondition-eml-spatialRaster
implementation-eml-software
individualName-eml-party
inline-eml-physical
inline-eml-resource
institution-eml-literature
institution-eml-literature
instrumentation-eml-methods
intellectualRights-eml-resource
interval-eml-attribute
issue-eml-literature
itemizedlist-eml-text
itemizedlist-eml-text

J

joinCondition-eml-constraint
journal-eml-literature

k

key-eml-constraint
key-eml-constraint
key-eml-constraint

key-eml-constraint
keyword-eml-resource
keywordSet-eml-resource
keywordThesaurus-eml-resource

L

language-eml-resource
language-eml-software
layout-eml-physical
lensDistortionInformationAvailability-eml-spatialRaster
license-eml-software
licenseURL-eml-software
lineNumber-eml-physical
lineNumber-eml-physical
listitem-eml-text
literalCharacter-eml-physical
literalCharacter-eml-physical
literalLayout-eml-text
lowWaveLength-eml-spatialRaster

M

machineProcessor-eml-software
maintenance-eml-dataset
maintenanceUpdateFrequency-eml-dataset
manuscript-eml-literature
map-eml-literature
maximum-eml-attribute
maximum-eml-attribute
maxRecordLength-eml-physical
measurementScale-eml-attribute
mediumDensity-eml-physical
mediumDensity-eml-resource
mediumDensityUnits-eml-physical
mediumDensityUnits-eml-resource
mediumFormat-eml-physical
mediumFormat-eml-resource
mediumName-eml-physical
mediumName-eml-resource
mediumNote-eml-physical
mediumNote-eml-resource
mediumVolume-eml-physical
mediumVolume-eml-resource

metadataProvider-eml-resource

method-eml-attribute

method-eml-entity

methods-eml-dataset

methods-eml-methods

methodStep-eml-methods

minimum-eml-attribute

minimum-eml-attribute

missingValueCode-eml-attribute

multiBand-eml-physical

N

name-eml-physical

name-eml-resource

name-eml-resource

name-eml-storedProcedure

nbands-eml-physical

nbits-eml-physical

nominal-eml-attribute

nonNumericDomain-eml-attribute

nonNumericDomain-eml-attribute

northBoundingCoordinate-eml-coverage

notNullConstraint-eml-constraint

numberOfBands-eml-spatialRaster

numberOfRecords-eml-dataTable

numberOfVolumes-eml-literature

numberOfVolumes-eml-literature

numberType-eml-attribute

numericDomain-eml-attribute

numericDomain-eml-attribute

numFooterLines-eml-physical

numHeaderLines-eml-physical

numPhysicalLinesPerRecord-eml-physical

O

objectName-eml-physical

offline-eml-physical

offline-eml-resource

offset-eml-spatialRaster

oldValue-eml-dataset

online-eml-physical

online-eml-resource

onlineUrl-eml-party
operatingSystem-eml-software
orderAttributeReference-eml-attribute
orderedlist-eml-text
orderedlist-eml-text
ordinal-eml-attribute
organizationName-eml-party
originalPublication-eml-literature
originator-eml-coverage
otherEntity-eml-dataset
otherEntity-eml-entity

P

pageRange-eml-literature
pageRange-eml-literature
para-eml-text
para-eml-text
para-eml-text
parameter-eml-physical
parameter-eml-resource
parameter-eml-spatialReference
parameter-eml-storedProcedure
parameterDefinition-eml-resource
parentOccurences-eml-constraint
party-eml-party
pattern-eml-attribute
peakResponse-eml-spatialRaster
performer-eml-literature
permission-eml-access
personalCommunication-eml-literature
personnel-eml-project
phone-eml-party
physical-eml-entity
physical-eml-physical
physicalLineDelimiter-eml-physical
pointInPixel-eml-spatialRaster
pointInPixel-eml-spatialRaster
positionName-eml-party
postalCode-eml-party
precision-eml-attribute
precision-eml-attribute
preProcessingTypeCode-eml-spatialRaster
presentation-eml-literature

primaryKey-eml-constraint
primeMeridian-eml-spatialReference
principal-eml-access
proceduralStep-eml-protocol
programmingLanguage-eml-software
projCoordSys-eml-spatialReference
project-eml-dataset
project-eml-software
projection-eml-spatialReference
projectionList-eml-spatialReference
protocol-eml
protocol-eml-methods
protocol-eml-protocol
pubDate-eml-resource
publicationPlace-eml-literature
publicationPlace-eml-literature
publicationPlace-eml-literature
publicationPlace-eml-literature
publicationPlace-eml-literature
publicationPlace-eml-literature
publisher-eml-dataset
publisher-eml-literature
publisher-eml-literature
publisher-eml-literature
publisher-eml-literature
publisher-eml-literature
publisher-eml-literature
publisher-eml-literature
pubPlace-eml-dataset
purpose-eml-dataset

Q

qualityControl-eml-methods
quantitativeAccuracyMethod-eml-spatialRaster
quantitativeAccuracyMethod-eml-spatialVector
quantitativeAccuracyReport-eml-spatialRaster
quantitativeAccuracyReport-eml-spatialVector
quantitativeAccuracyValue-eml-spatialRaster
quantitativeAccuracyValue-eml-spatialVector
quantitativeAttributeAccuracyAssessment-eml-attribute
queryStatement-eml-view
quoteCharacter-eml-physical
quoteCharacter-eml-physical

radiometricDataAvailability-eml-spatialRaster
rangeOfDates-eml-coverage
rasterOrigin-eml-spatialRaster
ratio-eml-attribute
recipient-eml-literature
recordDelimiter-eml-physical
referencedEntityId-eml-methods
referencedKey-eml-constraint
references-eml-resource
referenceType-eml-literature
relatedProject-eml-project
relationshipType-eml-constraint
repeats-eml-storedProcedure
report-eml-literature
reportNumber-eml-literature
repository-eml-coverage
reprintEdition-eml-literature
required-eml-storedProcedure
researchProject-eml-project
reviewedItem-eml-literature
role-eml-project
role-eml-resource
row-eml-spatialRaster
rowColumnOrientation-eml-physical
rows-eml-spatialRaster
runtimeMemoryUsage-eml-software

S

salutation-eml-party
sampling-eml-methods
samplingDescription-eml-methods
scale-eml-literature
scaleFactor-eml-spatialRaster
schemeName-eml-resource
section-eml-text
section-eml-text
sequenceIdentifier-eml-spatialRaster
series-eml-resource
shortName-eml-resource
simpleDelimited-eml-physical
singleDateTime-eml-coverage
size-eml-physical

size-eml-software
skipbytes-eml-physical
software-eml
software-eml-methods
software-eml-software
source-eml-attribute
source-eml-attribute
southBoundingCoordinate-eml-coverage
spatialRaster-eml-dataset
spatialRaster-eml-spatialRaster
spatialReference-eml-spatialRaster
spatialReference-eml-spatialReference
spatialReference-eml-spatialVector
spatialSamplingUnits-eml-methods
spatialVector-eml-dataset
spatialVector-eml-spatialVector
specimen-eml-coverage
spheroid-eml-spatialReference
standardUnit-eml-attribute
storageType-eml-attribute
storedProcedure-eml-dataset
storedProcedure-eml-storedProcedure
studyAreaDescription-eml-project
studyExtent-eml-methods
subscript-eml-text
subscript-eml-text
subStep-eml-methods
superscript-eml-text
superscript-eml-text
surName-eml-party

T

taxonomicClassification-eml-coverage
taxonomicClassification-eml-coverage
taxonomicCompleteness-eml-coverage
taxonomicCoverage-eml-coverage
taxonomicProcedures-eml-coverage
taxonomicSystem-eml-coverage
taxonRankName-eml-coverage
taxonRankValue-eml-coverage
temporalCoverage-eml-coverage
text-eml-text
textDelimited-eml-physical

textDomain-eml-attribute
textFixed-eml-physical
textFormat-eml-physical
thesis-eml-literature
time-eml-coverage
timeScaleAgeEstimate-eml-coverage
timeScaleAgeExplanation-eml-coverage
timeScaleAgeUncertainty-eml-coverage
timeScaleCitation-eml-coverage
timeScaleName-eml-coverage
title-eml-project
title-eml-resource
title-eml-text
toneGradation-eml-spatialRaster
topologyLevel-eml-spatialVector
totalFigures-eml-literature
totalFigures-eml-literature
totalPages-eml-literature
totalPages-eml-literature
totalPages-eml-literature
totalPages-eml-literature
totalPages-eml-literature
totalrowbytes-eml-physical
totalTables-eml-literature
totalTables-eml-literature
triangulationIndicator-eml-spatialRaster

U

uniqueKey-eml-constraint
unit-eml-attribute
unit-eml-attribute
unit-eml-spatialReference
unit-eml-spatialReference
url-eml-physical
url-eml-resource
userId-eml-party

V

value-eml-physical
value-eml-resource
valueAttributeReference-eml-attribute
version-eml-software

vertCoordSys-eml-spatialReference

verticalAccuracy-eml-spatialRaster

verticalAccuracy-eml-spatialVector

verticals-eml-spatialRaster

view-eml-dataset

view-eml-view

virtualMachine-eml-software

volume-eml-literature

volume-eml-literature

volume-eml-literature

vouchers-eml-coverage

W

waveLengthUnits-eml-spatialRaster

westBoundingCoordinate-eml-coverage

X

xCoordinate-eml-spatialRaster

xCoordinate-eml-spatialRaster

xIntercept-eml-spatialRaster

xSlope-eml-spatialRaster

Y

yCoordinate-eml-spatialRaster

yCoordinate-eml-spatialRaster

yIntercept-eml-spatialRaster

ySlope-eml-spatialRaster

Z