Ecological Informatics 36 (2016) 237–246

Contents lists available at ScienceDirect

Ecological Informatics

journal homepage: www.elsevier.com/locate/ecolinf

Ensuring the quality of data packages in the LTER network data management system

Margaret O'Brien [a,*], Duane Costa [b], Mark Servilla [b]

[a] Marine Science Institute, University of California, Santa Barbara, CA 93016, United States
[b] University of New Mexico, Albuquerque, NO 87131, United States

ARTICLE INFO

Article history:
Received 25 April 2016
Received in revised form 2 August 2016
Accepted 10 August 2016
Available online 14 August 2016

Keywords:
LTER network
Metadata quality
Ecological metadata language
Information management
Congruence

ABSTRACT

Considerable data analyses use automated workflows to ingest data from public repositories, and rely on data packages of high structural quality. The Long Term Ecological Research (LTER) Network now screens all packages entering its long-term archive to ensure completeness and quality, and to ascertain that metadata and data are structurally congruent, i.e., that the data typing and formats expressed in metadata agree with that found in data entities. The EML Congruence Checker (ECC) system is a component of the LTER Provenance Aware Synthesis Tracking Architecture (PASTA), and operates on data tables in packages described with Ecological Metadata Language using the EML Data Manager Library, written in Java. Checking is extensible for other data types and customizable via a template. Reports are retained as part of the submitted data package, and summaries here reflect the general usability of LTER data for a variety of purposes. On average in 2015, site-contributed data in the LTER catalog were 95% compliant (valid) with the current suite of checks.

1. Introduction

Data sets are an important contribution from Long Term Ecological Research (LTER) sites to the LTER Network Information System (NIS); these are intended to be used in cross-site synthesis projects for dissemination to federated catalogs and national repositories, and their long-term nature makes them irreplaceable for tracking environmental change (Gosz et al., 2010, Peters, 2010). In 2002, the LTER Network adopted an XML specification for its data exchange, Ecological Metadata Language or EML (Fegraus et al., 2005), followed closely by narrative guidelines for usage and recommendations for completeness (LTER, 2011). By mid-decade all LTER sites were contributing metadata records to a central catalog, and by 2009 were fully populating EML records (Michener et al., 2011, Porter, 2010). As with many other format specifications, an EML record supports machine reading and interpretation of its associated data entities, and code generators have been developed for ingestion of EML-described data entities into statistical, processing and database environments (e.g., Lin et al., 2008, Porter et al., 2012). The first-order quality standard for XML records is schema compliance, and for EML, additional parsing code checks that internal identifiers and their references adhere to specific rules (EML Project, 2008). However, experience with automated use of LTER data packages indicated that a significant fraction did not have metadata and data of sufficient structural detail (Leinfelder et al., 2008, 2010). Clearly, any automated use required a higher level of metadata and data quality and congruence, i.e., data typing and formats expressed in metadata must agree with that found in data entities. To assist data contributors as they prepare datasets, we developed a mechanism to provide feedback on congruence and potential usability to data contributors.

The Internet plus foresighted policies have fostered an enormous growth in the amount of scientific data available for download in all domains (e.g., AAAS, 2011). Further, the adoption of common, well-structured formats and metadata specifications allows for sophisticated machine reading. Some communities have developed conventions for specific data types or uses, usually as lists of defined fields with recommendations for use (e.g., CF Conventions, Eaton et al., 2011). But in many research domains, practices are still evolving for data's best handling and delivery. As has been the experience of the LTER Network, simple recommendations are inadequate, and a necessary continuation of these efforts will be benchmarks and compliance metrics.

Generally, assessments have focused on metadata content. NOAA (2014) has developed rubrics and metrics for metadata, which has been extended with summaries for specific use cases, e.g., for spatial data in GEOSS (Zabala et al., 2013). Habermann (2014) extends that approach further by evaluating XML metadata records using abstract metadata concepts (mapped to individual metadata specifications) against community-defined compliance levels and rubrics. The system has been implemented for some types of FGCD and ISO-19139

Abbreviations: EML, Ecological Metadata Language; LTER, Long Term Ecological Network; PASTA, Providence-Aware Synthesis Tracking Architecture; NIS, Network Information System; ECC, EML Congruence Checker; CF Conventions, Climate and Forecasting Conventions; GEOSS, Global Earth Observation System of Systems; IMC, Information Management Committee; DML, Data Manager Library; THREDDS, Thematic Real-time Environmental Distributed Data Services; netCDF, Network Common Data Form.

* Corresponding author.
E-mail addresses: margaret.obrien@ucsb.edu (M. O'Brien), dcosta@unm.edu (D. Costa), servilla@unm.edu (M. Servilla).

http://dx.doi.org/10.1016/j.ecoinf.2016.08.001
1574-9541/

documents, with results aggregated into summaries for review. For Linked Open Data (LOD) the "LOD Laundromat" (Beek et al., 2014) converts idiosyncratic input to a "cleaned sibling" (their term), and so removes the contributor from the cleansing process. The set of heuristics applied is mainly syntactic, with semantic interpretation of content to detect duplicate triples. Results can be aggregated into bulk reports on input quality. LTER efforts presented here represent an intermediate approach: we examine metadata from one specification (EML), data entities therein, plus their agreement (termed "congruence"). This strategy means we can examine datasets more deeply than with the Habermann approach, but we do not attempt to correct metadata (or data) per the LOD Laundromat. The reasoning is two-fold: first, LTER needed to assure more than just presence of metadata elements, it was important to assure that data entities could be machine read, thus the need to examine congruence. Secondly, most ecological data are so complex that heuristics for their repair were simply beyond the scope of this project. Hence, our system informs submitters of its findings for their judgment and repair.

The LTER Network has developed the EML Congruence Checker (ECC) to inform the dataset contributor about the structure of the data package, and indicate whether the asserted metadata accurately defines the data entity (table), i.e. to ensure "structural congruence". There is minimal semantic checking. The ECC was developed with considerable community involvement, concomitant with the development of other advanced software for the LTER NIS (Servilla et al., 2016). All code and schemas are open source, and the system has been in production since 2013. Today, every incoming data package is subjected to up to 32 distinct checks, which encompass a variety of data and metadata features ranging from simple confirmation that certain metadata XPaths are present to assurance of congruence between metadata and data.

Approximately 60 checks are awaiting consideration, and implementation continues within other management constraints. The system has the potential to produce additional descriptive material for data values themselves, which may be developed at some later date, e.g., value ranges, frequency distributions, and qualitative comparisons to metadata content. We include here summaries for reports to date, and discuss error modes, highlighting ways that reports may provide input to the design of specific tools, or help identify gaps in a data management system.

2. Methods

2.1. Community input

An initial outline for data package checking was constructed in 2009 by a group of LTER site data managers and NIS developers, plus representatives from the National Center for Ecological Analysis and Synthesis (NCEAS), a major partner in the development of EML and its associated code. The ECC was developed as an LTER product, but throughout the process a broader community of data practitioners was engaged to ensure the ECC would be widely useable, with progress reports and/or breakouts formed at national meetings, e.g., the Environmental Information Management Conference in 2011 (http://eimc.ecoinformatics.org) and to the American Geophysical Union (AGU, 2013). A working group of LTER site data managers was convened to provide overall guidance of the ECC development. Communication with LTER site data managers took several forms: progress was presented at regular annual meetings of the LTER Information Management Committee (IMC) and with written reports. Meeting breakouts, ad hoc virtual meetings or independent workshops were formed to advance specific objectives. Communication with NIS developers used the Tiger Team mechanism established for other NIS components (Servilla et al., 2016). Participation in all activities was voluntary.

In 2010, the IMC working group, NIS developers and Data Manager Tiger Team began compiling a list of potential checks, and in 2011, five were implemented and tested against thousands of LTER data packages. To finalize the comprehensive set of checks in Spring 2012, we convened a workshop for a small group of community experts from within and outside LTER with extensive knowledge of EML and the issues exhibited by existing data packages. The 2012 workshop participants were asked to:

A. Determine specifically what quality checks would be required to meet the criteria of the LTER community for high quality data packages, guided by established narrative Best Practices (LTER, 2011)
B. In addition to LTER Best Practice recommendations, consider EML construction currently in use in LTER metadata contributions
C. Prioritize checks for the greatest return on investment

Several system features were specifically requested to allow the system to evolve and be adapted for a variety of uses, and to facilitate adoption by data contributors.

1. ECC should accommodate the addition of new checks and staged implementation
2. Configuration should be customizable for use by different communities
3. Checks which will return "error" (and prevent insertion of a data package) should be implemented first to highlight the most important issues
4. Code should operate in two modes
   a. "Evaluate", in which checking continues after a failure so that a submitter sees as many problems as possible all at once
   b. "Harvest", in which checking stops on the first error and the data package is rejected, and results are returned as quickly as possible
5. An HTML interface should transform XML report results for easy viewing

2.2. EML data manager library

In addition to requirements set by the community, the system architecture would take advantage of existing software. The EML Data Manager Library (DML) is part of the EML family of tools, and was designed as a software library for parsing EML metadata, creating relational database structures from entity metadata, and loading the associated data into the resultant database, supporting query and selection operations on that data (Fig. 1) (Leinfelder et al., 2008, 2010). It includes an Application Programming Interface (API) for interactions with the library. Although the initial implementation of the DML was very efficient at automating metadata parsing, data loading, and database query operations, it lacked a structured means of evaluating or reporting on the quality and congruency of the metadata and data.

2.3. PASTA integration

During the early stages of PASTA's development, it was recognized that the DML had the potential to play an important role in an automated process of assessing the suitability of candidate data sets for inclusion in the PASTA repository. To accomplish this, however, the DML would need to be extended with a suite of formal checks - data structures and procedures that furnish it with the means to perform evaluations on the metadata and data it traverses, together with the means to report on the outcomes of those evaluations. Once implemented, this added layer of functionality effectively transformed the DML from a "black box" to a "glass box", a quality engine that could serve not only as a gatekeeper for PASTA data set ingestion but could also explain its rationale for either accepting or rejecting data sets. These functions are contained within the PASTA Quality Engine (Fig. 2). A full description of the PASTA architecture is beyond the scope of this paper, and readers are directed to Servilla et al. (2016).

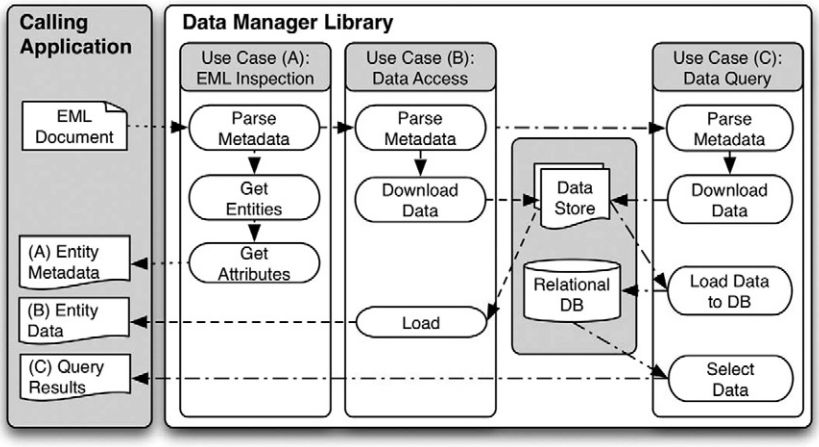

Fig. 1. EML Data Manager Library (from Leinfelder et al., 2010), showing three use cases. An application requests information from the metadata (A), downloads the data to the host data store (B, C), and creates backing tables in an associated relational database (C).

3. Results

The history of development is presented in Fig. 3. By Spring 2012, a total of 72 checks had been proposed, and many finalized during the workshop itself. Checks are continually maintained in a collaborative online document and archived as needed (Dataset: O'Brien et al., 2016). At the time of this writing, the total stands at 91 checks entered, with 32 implemented, and 21 designated as "deprecated" or

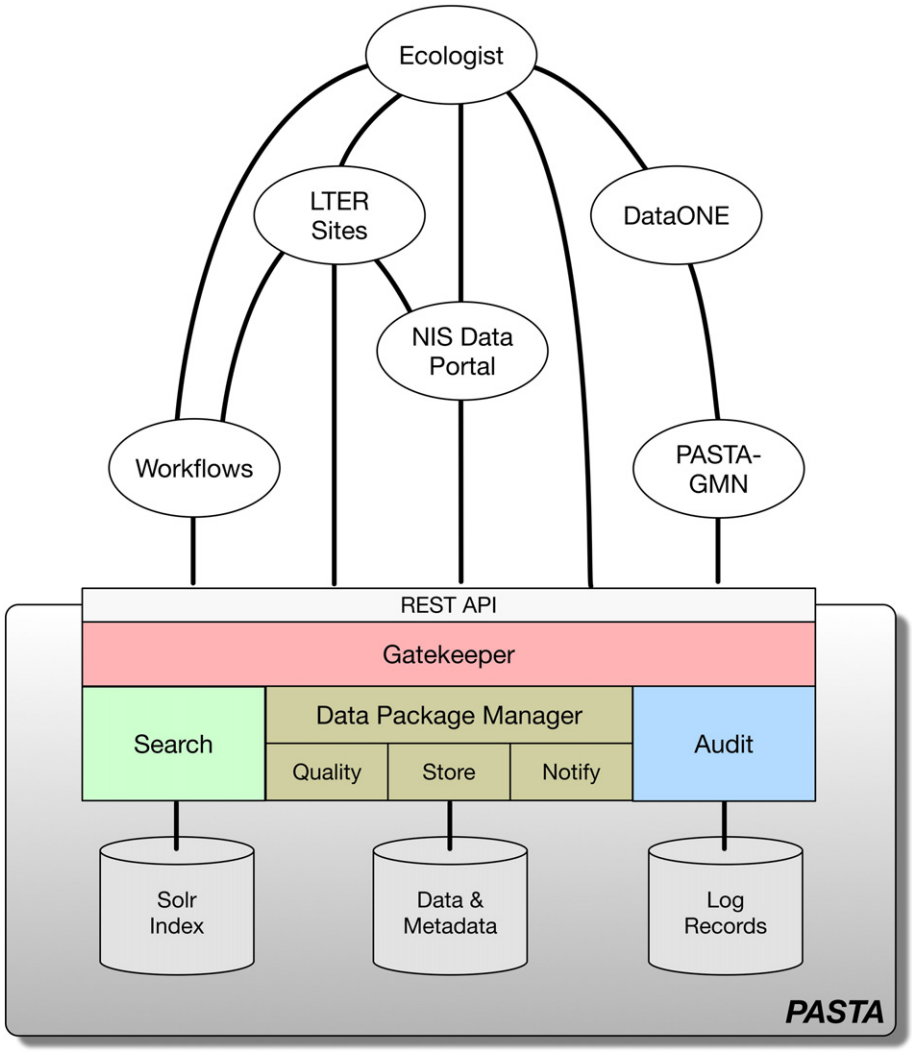

Fig. 2. Conceptual diagram of the PASTA service stack and a generalized set of ecological data producers and consumers (reproduced from Servilla et al., 2016) LTER sites contribute data packages via the REST APIto the Data Package Manager, where they are evaluated against the checks.

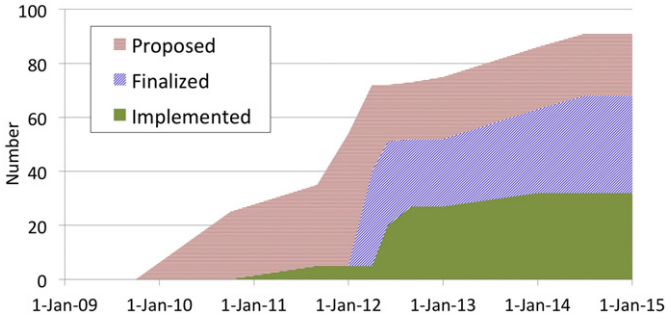

Fig. 3. History of ECC development. Significant events include (1) initial discussions at LTER All Scientists Meeting, (2) Proof of concept released, (3) Workshop to finalize system requirements and majority of check descriptions, (4) PASTA release.

"postponed". Deprecated checks were sometimes obviated by other entries, but their record retained to avoid future duplication. "Postponed" checks are generally too complex to be addressed quickly, e.g., reflecting complex features of data values themselves that may require additional definition and coding (e.g., allowable ranges, or conformance with external vocabularies and formats).

3.1. Checks described and categorized

In addition to basic pass/fail criteria, each check's definition includes categorization according to several features: scope, justification, response behavior, packaging aspect, and priority. Although some typologies simply facilitate organization or communication (e.g., justification, priority), having a specific, granular definition for each check forced contributors to focus on their most salient features and facilitated coding. Checks that would prevent insertion were considered and justified with special care. The high number of checks recorded to date (91) reflects the complexity of datasets submitted, and the granularity allowed by EML metadata.

Scope: The new capability added to the DML can easily be used by others with systems based on EML. Because communities are expected to employ different criteria for data package acceptance, each check can be categorized with a 'scope', to indicate the community applying it. General checks that are likely to apply to any data package (e.g., presence of working URLs) were given the scope "knb", to signify the "Knowledge Network for Biocomplexity", the parent project for EML development (EML Project, 2008). Checks that are specific to the LTER community are labeled "lter", e.g., features recommended by LTER EML Best Practices (LTER, 2011). Other values for 'scope' may be added by interested communities.

Justification: The LTER IMC specified that data package checking should not cause undue burden for data package submitters, and so the value of each check must be justified. "Discovery" justification applies to those elements used by search tools or during human evaluation. "Workflow" was applied to checks for data package features essential to workflow software and automated ingestion. "PASTA" refers to data package features specifically required by core NIS software components (Servilla et al., 2016). "Good practice" was gleaned from EML Best Practices documents (LTER, 2011), the EML specification and documentation, and requests from LTER governance or NSF. Experience indicates that simple written recommendations are insufficient to encourage the adoption of good practice, and that a mechanism for reporting on compliance could help. In some cases, a check may belong to more than one category; for example, a "good practice" may have been defined as such to promote "discovery". In those cases, the more explicit justification prevails.

Response behavior: Central to code behavior is its response to a check; and some checks affect the insertion of the package into the NIS. There are a total of four possible responses ("info", "valid", "warn", or "error"), in two major classes. Checks designated as "info" do not have pass/fail criteria and do not affect the acceptance of the data package in any way. An "info" check is for informational purposes only; for example, the check to display a few lines of content of a URL.

The second class of check can affect insertion of a package into the NIS. Their responses will be either "valid", or one of "warn" or "error". "Valid" means that all criteria of the check were met. An "error" response during the checking process will cause the entire data package to be rejected. "Warn" means that the criteria of the check were not met and that there may be some problem needing attention, but that the data package is still acceptable. With two levels of non-valid response ("warn" or "error"), code behavior can be customized (see below).

Understandably, checks classified with the "valid/error" response behavior were of greatest importance to classify correctly, because these would deny upload to PASTA. Only checks whose failure would mean that a data package is unusable should generate an "error". These include checks for:

(a) XML documents that do not comply with the EML schema because these cannot be transformed to HTML or their XPaths interpreted
(b) Package identifiers outside the controlled LTER Network format, as these cannot be entered into Network catalogs
(c) Metadata URLs for data entities that are broken links, because data cannot be accessed by any means
(d) Non-unique entity names in metadata (within one package), because individual data entities cannot be distinguished
(e) The count of entity attributes (e.g., columns) in metadata does not match the count of columns in data entities, because incongruity generally means (at best) jagged rows, which is unacceptable to analysis environments like R or Matlab; or (at worst) the metadata does not belong with this data entity, which renders the package unusable.

Packaging aspect reflects the part of a data package where a check operates. "Metadata" checks are concerned purely with metadata presence or content, e.g., a check for the presence of an XML element, such as "<methods>" or "<geographicCoverage>". "Data" checks are concerned only with the data entity, e.g., a check that simply returns a row of data or examines a data record for possible delimiters. "Congruence" checks examine the agreement between metadata and data, e.g., to compare the number of attributes listed in metadata to the number of columns in a data table.

Priority: Each check was given one of three priority levels (high, medium, low) depending on its importance to LTER and to the IMC. Priority levels may help guide the implementation, but are not the only factor used to determine the schedule.

Table 1 shows the distribution of checks among three typologies that were either essential to code development, or are of particular interest to the community: "Packaging aspect", "Response behavior" and "Justification". For "Packaging aspect" about ⅔ of checks belong to the "Metadata" type, meaning that they pertain to basic metadata content. Most of those that simply examine presence of an element and follow the same coding pattern have been implemented (40%). Very little analysis of metadata content (e.g., semantics) has been attempted to date, but the framework is flexible enough for additions. For example, a check is implemented for the presence of a "<keyword>" element, with a response behavior of "valid/warn". A related check has been proposed to specifically check for terms from the LTER Controlled Vocabulary (LTER, 2013), and with a corpus as diverse as LTER's, one can imagine that connections to other community vocabularies will be requested as well. These sorts of semantic checks will entail loading of external resources, which was out of scope for the current system. Knowing their importance and classification helps programmers to plan for their future incorporation.

A considerable number of "Congruence" checks are not implemented (70%); this is because overall, these are the most complex, requiring

Table 1
Distribution of checks within three typologies, and the proportion of entered and typed checks that are implemented. As of late 2015, 32 checks have been implemented; of the remaining 59 entered, some have not yet been classified in all typologies.

Typology	Type	Entered	Implemented	
			Number	Proportion
Packaging aspect	Metadata	58	23	40%
	Data	6	2	33%
	Congruence	23	7	30%
Response behavior	Valid/error	20	12	60%
	Valid/warn	32	13	41%
	Information	21	7	33%
Justification	Workflow	31	14	45%
	Best practice	36	10	28%
	PASTA system	9	2	22%
	Discovery	7	6	86%

ingestion and analysis of entire data tables. However, this group is highly anticipated by data managers and scientists, as they include descriptive and statistical displays of data values themselves, and will ensure more streamlined usability by workflows. As an example, four individual congruence checks are required to ascertain that a data table is not "jagged", i.e., that all rows have the same number of fields. Uneven rows are a fairly common occurrence in tables exported from spreadsheets. The checks operate as follows:

1. The record delimiter is examined and compared to metadata ("valid/warn", since default delimiters can be safely assumed).
2. The field delimiter in metadata is examined and deemed acceptable ("valid/error")
3. Each data line is parsed, fields counted and compared to an expected number (ascertained by counting table attribute metadata nodes). Hence, the final two checks are "too few fields", and "too many fields"; each is classified "valid/error".

Every suite of checks at this level of detail requires significant interaction with data providers to ensure the appropriate check settings and reasonable code behavior.

Per IMC requests, overall the highest proportion of implemented checks is in the "Valid/error" type (60%), as these have the most consequences for submitters. The 11 implemented "Valid/error" checks detect the aspects of usability described above. The nine remaining candidates pertain to planned functionalities such as delivery of archives and integrity checking via checksums, or to potential usage of external code sets or inter-entity relationships. Their definitions and implementation are likely to require significant further discussion. 41% of the "Valid/warn" type have also been implemented, but 90% of these were implemented before January 2013, which means that submitters have received consistent reporting of their packages since the NIS went into production.

A requirement of the system was that it should promote good practice by attempting to detect violations of Network or EML-community recommendations (LTER, 2011). These checks are justified under the "Best practice" type, and reflect the highest proportion of checks under the "Justification" typology. Many of these checks are for the presence of certain metadata elements (e.g., "<geographicCoverage>"), and so often are co-classified with the "Metadata" checks (see dataset, O'Brien et al., 2016). An increasingly common use of EML-described data is ingestion by workflows and automated systems, and checks that pertain to these entity and metadata features are classified as "Workflow"- justified. Because ingest-scripts are built from EML-metadata (Lin et al., 2008; Porter et al., 2012), these checks are generally concerned with how accurately metadata describes the data entity, because structural features such as delimiter and line ending often must be specified for ingest into some analysis environments (e.g., R, Matlab), rather than being automatically detected as with spreadsheet ingestion. Hence, a high proportion of "Workflow" checks are also "Congruence" checks (Dataset: O'Brien et al., 2016).

The use of multiple typologies means that checks can be tuned for specific cases. For example, EML contains an element "<coverage>", with three children for geographic, temporal and taxonomic coverage. A check for the parent element "<coverage>" is set to a response of "warn" because the community felt that every dataset should be accompanied by at least one coverage element. But the checking system cannot ascertain the scientific domain of a dataset; so it not capable of choosing which child nodes should be required. Hence the checks for child-elements (taxonomic-, temporal- or geographicCoverage) carry a response of "info".

3.2. Data package quality report

An XML schema was designed to contain the output of the Quality Engine containing a description of each check (Fig. 4), and an instance document is called a Data Package Quality Report. Using XML to house the report allows output can be transformed for a variety of purposes, e.g., an individual report can be transformed into HTML for web presentation during evaluation of a single data package, or results from a group of reports can be aggregated and statistics computed.

The expectation is that different communities will want to control the behavior of some checks. So every check can be configured in an XML template (an instance document of the report schema), and the XML template then controls the behavior of the checker code.

3.3. Code behavior

A NIS Data Package Manager Web service component called the "Quality Engine" codifies checks during data package analysis (Fig. 2),

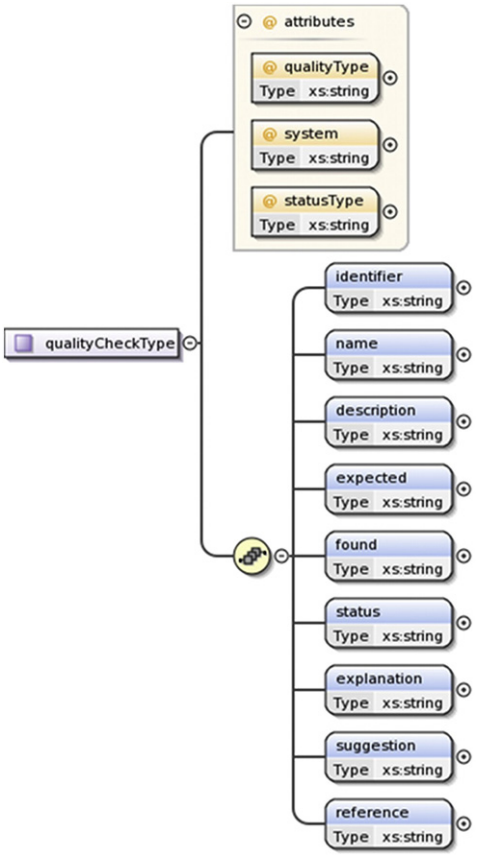

Fig. 4. Report XSD for a qualityCheckType in a quality report. A qualityCheckType may appear in report XML at both the dataset and entity levels (not shown).

applying the XML template for each check. It can be run in two modes, "Evaluation", for pre-submission checking, and "Harvest" mode for controlling data contributions to the catalog. Development of an evaluation-mode was specifically tailored to data submitters' work patterns.

3.3.1. "Evaluation" mode

Typically, software for evaluating XML stops at the first error and repeated submissions are required until all errors have been exposed. The community requested that, at much as possible, all errors be exposed in one run (rather than stopping processing) so that a submitter sees as many problems as possible at once. This feature will save package submitters considerable time. Of course, some errors will prevent future processing, e.g., if a data-URL does not return a data entity, that entity's delimiters cannot be examined. Reports from Evaluation mode are stored for 180 days, and made available to submitters via the upload interface.

3.3.2. "Harvest" mode

When run in harvest-mode, the Quality Engine will halt on the first error, thus saving processor time. Upon a successful harvest, the quality report document is stored permanently as part of the data package (associated via the resource map), and can be accessed and displayed alongside its metadata and data. All LTER data in the network data portal datasets are now accompanied by a quality report; readers wishing to peruse reports are directed there.

3.4. Report results

An overview of data packages submitted to the LTER catalog can be found in Servilla et al. (2016). Here, we will summarize results of ECC reports from 9353 data packages from 28 submitters over a three-year period, 2013–2015. For the most part, submitters are the current 25 LTER sites, but also included are ad hoc packages contributed by the Network, and packages from former LTER sites whose data are still attributed to that project, but managed by centralized staff. Report data were gleaned from publicly available reports accompanying each data package (Dataset: Costa, 2016).

Reports for datasets uploaded do not contain responses for "error"-type checks because packages with errors were denied with no public reports retained. Hence, our analysis here is limited to the 13 checks that return "warn". "Warn" checks are of particular interest because they represent dataset features that exhibit minor problems or deficiencies that the community felt would be likely to be resolved by the submitters had they been made aware. Every warn-check was hit by at least one submitter. The overall average rate was 5% (total warns/total checks run), indicating that generally, submitters are capable of producing valid packages (Dataset: Costa, 2016). Report data were analyzed in two ways: by check, and by submitter (anonymous). Results for warn frequency (number of submitters affected) and warn rate (warns/data package, %) for 13 checks are summarized in Table 2, grouped by "Justification" (Workflow, Discovery and Best Practice), and Fig. 5 shows the warn rate by submitter cumulatively (A), and over time (B).

3.4.1 By check, justification typology

3.4.1.1. Workflow. Because of the importance of the ability to create ingestion scripts from EML metadata, 10 of 14 implemented "Workflow" checks were classified to return an "error" on failure, and so deny admission to PASTA. All four of the remaining checks classified as "warn" were seen in submitted data packages. The highest frequency of "warns" in any category was seen for the check that ascertains whether a data table can be loaded into a relational database (RDB) table via its metadata (Table 2, "Workflow"). RDB-loading was chosen as the implementation method because generally, data definition languages are strict about typing and formats, and so if metadata was sufficient to create a relational database table and the entity loaded, then quite probably, that entity is structurally congruent with its metadata and adequate for ingestion to other software as well. The importance of being able to load any data entity into an RDB is debatable, and so this check was classified as "warn". However, the community requested that checks be planned for range-checking or summary statistics for the data values themselves, plus reports of congruence between data values and metadata (e.g., comparing date ranges in data to coverage dates in metadata). To accomplish those more advanced checks, data values must be analyzed, and having an early report on RDB load-status informs the community about the feasibility of running more complex checks on the data they submit. Hence, Table 2 shows that for 12% of the data from 24 submitters (almost 90% of the network), these informative reports currently could not have been produced.

The second ranking check (17 submitters affected, Table 2) is concerned with automated detection of record delimiters. EML allows submitters to specify the record delimiter with an optional element for each entity, which is not essential because some software (e.g., spreadsheet) is able to detect record delimiters on import. Only two submitters did not include a <recordDelimiter> element in at least one package. However some software may produce artifacts unless the record delimiter is understood correctly during import (e.g., interspersed empty rows in R). Because LTER data management systems receive data from a variety of sources, their data entities are rarely uniform, and this check was designed to inform submitters when their data had line endings that did not match expectations, and might cause problems for ingestion software. It is impossible for a single automated checking system to mimic the

Table 2
Checks for which any package received a "warn", by justification-type, with frequency (number of submitters having one or more warns for that check) and the total warn rate among affected submitters.

Justification	Check description	Warn frequency # of submitters	Warn rate (%)
Workflow	Data can be loaded into a PostgreSQL table using entity-level metadata.	24	11.98
	Data are examined to detect possible record delimiters (other than that specified in the element "<recordDelimiter>").	11	14.06
	Record delimiter element is present ("<recordDelimiter>").	2	7.65
	Attribute names are unique within a data entity.	8	1.32
Discovery	Dataset abstract element is a minimum of 20 words.	18	8.21
	Methods element ("<methods>") is present.	17	8.64
	Dataset title length is at least 5 words.	12	2.64
	An entity description is present.	9	9.34
	At least one keyword element is present.	6	2.47
	At least one coverage element is present (EML allows three: for temporal, geographic and taxonomic coverage).	5	10.76
Good practice	Number of records in metadata element "<numberOfRecords>" matches number of records loaded into database table.	8	0.56
	Length of entity name is not excessive (content of "<entityName>").	7	1.49
	<pubDate> element is present.	5	23.96

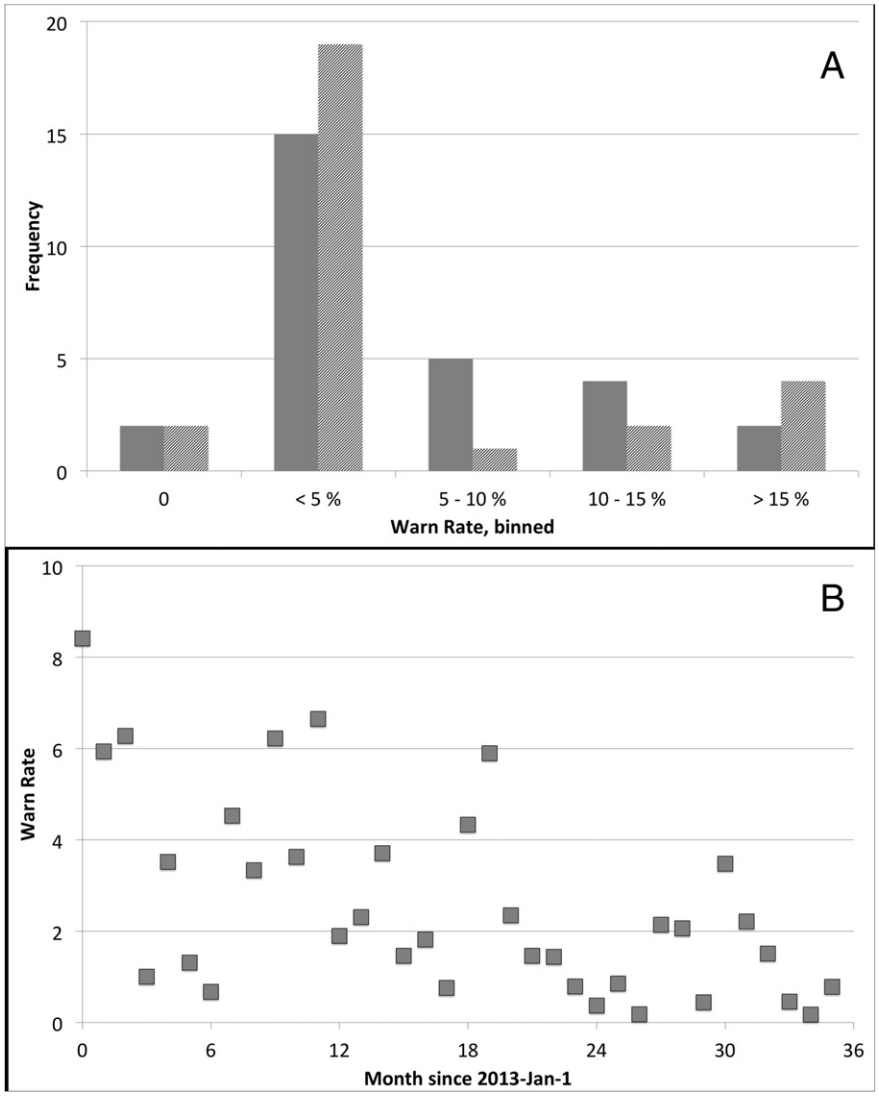

Fig. 5. Warn-rate for 28 data package submitters, 2013–3015, where warn-rate = percent warns/total checks run that are classified as "warn". A. Frequency: solid bars, warn rate for all 13 checks; crosshatched bars, for 7 entity-level checks. B. Warn-rate across all submitters, aggregated by month since January 2013.

behavior of every coding environment, but these reports indicate that 8.6% of data from 17 submitters might have difficulty being imported into an analysis system.

The final check in the "Workflow" group ascertains that attribute names are unique with an entity. All analysis software uses string variable names, and the variable names must be unique, a stipulation also required by database tables. Even in spreadsheets, users would be confused with multiple columns having the same name. So this check was constructed to alert submitters when their data entities were constructed with non-unique attribute names. Only eight submitters saw this warning for any packages, and at a fairly low rate, 1.3%.

3.4.1.2. Discovery. "Discovery" is the ability of a user to locate a data package in a catalog via its metadata, and ascertain that this indeed is the correct data. Often, only high-level metadata about context, descriptions, coverage or timing are useful for this purpose. Many of these metadata elements are text nodes. Dataset citation is a relatively new practice, and also a form of advertisement or discovery, and citations also use high-level text nodes. In general, although text nodes may contain valuable contextual information, little of the actual content can be analyzed within this automated checking software; checks that attempt to mimic even minimal editorial or semantic review are out of scope. All "Discovery" checks are classified as "warn".

We were able, however, to design checks that examined some basic features of text nodes. Observations of data submitted by LTER sites to the first-generation catalog showed that some used a simple file name (which may be quite cryptic) as a dataset "title", possibly as a placeholder early in the development of an IM system. Because dataset titles are part of the citation, a check was constructed to examine a basic aspect - the number of individual words. We examined all existing datasets titles in the first generation LTER catalog and found the distribution to be bimodal: titles were either a single word in length (the aforementioned file name), or ranged from 7 to 20 words. So five words was established as the minimum acceptable title length. A similar analysis was carried out for dataset abstracts and entity descriptions.

Highest frequencies were observed for short abstract (fewer than 20 words) and an absence of an explicit methods tree. Both of those text elements help users to learn about the data's context, and ascertain if data are appropriate for their use, so problems with these may mean data are less than fully usable. However, only ~1% of packages from eight submitters had both a short abstract and no methods, which may mean that submitters put all the important contextual information into just one or the other of these two nodes. This may be

adequate for manual evaluation, but would mean that package design is inconsistent across the Network, or perhaps even within a submitter.

3.4.1.3. Best practice. The "Best practice" group has the highest number checks in the "Justification" typology (13, Table 1), but about half are classified as for "information" only. Three checks were deemed important enough to return a "warn" on failure. The highest warn rate (Tables 2, 23% packages from 5 submitters) was found for the optional element "<pubDate>", whose content is used to build dataset citations. As dataset citations become more important, it may be valuable for the Network to address this issue, so that complete citations can be constructed for all packages. A check for excessive length of content in data entity names leaf nodes (a string type, "<entityName>") was designed to highlight content that might be better placed in textType elements, and "warns" were returned at a fairly low rate (1.5% of packages from 7 submitters). Finally, a check to compare the asserted number of rows (in EML metadata) to the number of rows loaded into the database is an example of a potentially useful check that might be underused. As the EML element is optional, the check is not even run unless included. Overall, this check was run on fewer than half of all entities (~12,000, or 42%) distributed across all submitters. But it returned "warn" at a very low rate (0.5%) from only 8 submitters, showing that when it is used, the check can act as a confirmation that all data uploaded can indeed be read, and that all expected data are present.

3.4.2. By Submitter

Fig. 5a (solid bars) shows the frequency of overall warn rate (number of warns/all checks with warns) across all submitters. Two submitters had no warnings at all (warn rate = 0), and the maximum warn rate was 16.7%. More than half the submitters (15) had overall warn rates <5%, and only a handful have rates over 10%. Differences in data packaging patterns can affect the overall warn rate, and packages with many entities will have a higher average warn rate than if the packaging cardinality is 1:1 (entity:package). To remove that effect, we examined the warn rate only for the seven checks which apply to data entities (Fig. 5a, crosshatched bars). The general pattern was the same: most submitters have warn rates under 5%. There is a slight increase in the highest bin (> 15% warn rate) when only entity-level checks are considered, which may mean that some submitters have more problems creating data tables suitable for workflow ingestion than they do with high-level metadata for discovery.

Reports on thousands of packages accumulated over the three years since PASTA roll-out means that there is sufficient data for reports to be examined for a change in the rate of warnings over time. For the period January 2013 through December 2015, about 60% of packages were uploaded with no warns, and the overall trend has been toward a lower number of warns per package (Fig. 5b); this is also the trend for most individual submitters (data not shown).

4. Discussion

4.1. New checks and future development

The system will accommodate new checks as the community determines these are necessary, and Fig. 3 shows 19 checks added since initial implementation in 2013. Further, existing checks may require modification; for example, a check with a response status of "*warn*" in 2013 may require reclassification to return an "*error*", or conversely a check's response status may be relaxed. The declarative format of the template allows such changes to made with ease; simple edits to the XML file can modify a check's classification without the necessity of recompiling and redeploying the DML's underlying Java code. However, it is imperative that changes to the check configuration be implemented with adequate notice to submitters, and without causing undue burden. Currently, new checks are reviewed by the original IMC working group and implemented as part of regular PASTA maintenance, which has been adequate for the small number to date. However, we anticipate a more formal process to be defined in the future, where checks are reviewed periodically and changes or modifications announced, with an appropriate comment period and schedules for implementation.

Possibly, as more new checks are implemented, patterns will emerge for how best to handle certain cases. For example, it should be obvious that if a check is not run, it can make no assertion about the data. The check validating the number of asserted records runs on an optional element ("<numberOfRecords>"), and so currently is not run if the element is not included in metadata. However if including the element is indeed "best practice", then a check ought to first report on its presence, followed by congruence with data - similar to the pair of checks that first look for the presence of a record delimiter, and then examine records for other possible delimiters.

A considerable number of proposed checks remain to be implemented. Future work continues to be guided by community priorities and pragmatic coding decisions (e.g., check patterns, EDL code limitations and time), two features that make check development a somewhat gradual process. Communication with a broad community of providers is absolutely essential - beyond the fact that user involvement fosters acceptance - as they hold the in-depth knowledge of data's context, usage and issues. Further, the check typologies were essential to elucidate patterns, and ensure appropriate settings and reasonable code behavior, and typologies could not have been developed without a comprehensive initial list of checks. The LTER Network already had a basis for metadata quality to provide considerable initial material in its narrative Best Practices guide, which is granular and focused on the EML specification. In a broader context, e.g., that of a metadata aggregator handling multiple specifications, establishing that basis is still important, but is likely to be challenging (Tania et al., 2013).

4.2. Effect on behavior of data submitters

The ECC has been in place since early 2013, and over the subsequent three years, an overall decrease in the warn rate has been detected in data packages (Fig. 5b). However, to more fully understand submitter behavior and the effort required to produce a non-error or non-warn data package, more could be learned by examining reports from the evaluation mode, which is run before submission. We present no reports from the evaluation mode and focused only on reports of uploaded packages. A more thorough analysis would be required to answer questions such as "*for those data package that have multiple revisions, is there a pattern of improvement?*" by examining patterns of both warns and errors and following them through repair and upload. Because evaluation reports are not public and stored for only a few months, this activity should be collaboration between submitters and the Network.

4.3. Uses of reports

A formal system of check management would be complemented by a centralized reporting mechanism in which all data packages are examined and their compliance with current checks summarized. The IMC has already formed a working group to define appropriate content and reporting practices for a variety of audiences, e.g., data managers, site PIs, LTER governance, or NSF. It was also envisioned that checks could be used to guide development of data management tools. In the most general scenario, an individual site may be requesting supplemental funding for its local data management system, and could use the reports as part of their proposal justification. For example, generally poor high-level metadata may indicate the need to adopt or update a central relational database system to more easily control content at origin.

Ecological data generally fall into the "long-tail" of data type-distributions, in that there is a small quantity of each type. Further,

ecological data are frequently handled manually in spreadsheets or with ad hoc code. So unlike sensor data, where a few instrument configurations control most of the output and data are relatively straightforward to characterize, ecological data sets tend to be unique. However, their handling could be made more efficient if common problems were identified and targeted, and general-purpose software solutions developed for them. Ideally, the process for tool development should not be based on laboratory-specific anecdotes, but instead on rigorous analysis of a significant corpus of ecological data. LTER can supply that corpus, and code such as the ECC could form the basis of that analysis.

Many online data sets are now available via URLs with sophisticated metadata, e.g., physical sensor data in THREDDS or NetCDF. Researchers would like to be able to integrate data from these services with ecological data - especially with time-series such as those produced by the LTER. However, given the diverse nature of biological data, this is often a difficult and manual process. Currently, about 90% of submitters encounter "warn" on one or more of the checks associated with automated usage (Workflow typology, Table 2). The Network could follow up on these warnings to find their root cause, and plan for ingestion software to circumvent the specific problem, e.g., a data table ingestion system that standardizes the line endings and generates appropriate metadata content. Such services could alleviate many of the structural problems, making LTER data more usable for many types of automated ingestion.

A mature data system responds to metrics; however, these cannot be constructed without some basis. Report content can be used to develop maturity levels, and then track progress through them via metrics. Our first order analyses here focus mainly on frequencies and rates of warns. A more thorough analysis should consider the different typologies, for example, to distinguish between checks that target data entities vs. those for high-level metadata (e.g., per Fig. 5a). Overall ratings of the usability of a data package for specific purposes could be constructed from groups of checks. Broadly scoped initiatives include the Stewardship Maturity Matrix (Peng et al., 2015) and AGU's Data Management Maturity service (American Geophysical Union, 2013), which are constructing vocabularies and measurements of data management practices and data products to improve consistency and enhance discoverability and reuse. Format-specific evaluation tools such as the ECC complement those efforts.

4.4. Limitations

4.4.1. Entity types supported

The EML schema specifies nodes for six types of data entities, and the Quality Engine can process aspects of "dataTable", "spatialRaster", "spatialVector", and "otherEntity" entities. Programming logic has not yet been developed and tested to process "storedProcedure" and "view" entities. Most entity-level checks apply to "dataTable", and within the entity type "dataTable", only simple delimited formats are currently supported by the Quality Engine. In the LTER, the vast majority of data entities are indeed dataTables, however, as local systems become more sophisticated, the need to developing checking criteria for other types will increase.

4.4.2. Data manager library

The choice to use a relational database in the data manager library may not be appropriate for all checks, even those focused on tables, and some limitations were encountered, e.g., with date-time typing. In fact, the high warn-frequency for loading an entity into a relational database table (Table 2) may be due in part to these limitations. However, as the ECC is a community project, implementation can be reviewed with a broad, well-informed community, and suitable solutions found.

5. Conclusion

Overall, uniformity and usability of datasets in the PASTA catalog is higher than what was found in the first generation LTER data catalog. We cannot quantify that improvement because experiences recorded with the first-generation catalog were anecdotal, and any checking was ad hoc or cursory at best. However, simply setting rules for admission to a catalog - with the involvement of the contributing community - has improved the landscape by encouraging redesign of data packaging systems, and helped to persuade the adoption of consistent, good practice. Further, we have shown here that with a rigorous system such as the LTER has with the PASTA Quality Engine, we can track progress and improvements quantitatively.

Not all aspects of a dataset can be checked automatically; some still require human evaluation or editorial review (e.g., abstract text). However, an automated checker is able to make the job easier, and to highlight areas of community concern. It is still possible to upload data to PASTA with imperfect metadata, but overall, awareness has risen, and content is more uniform and complete. Common practices (and some poor practices) are now visible, and submitters appear to have the ability to respond to feedback quickly. With the ECC, these data set features can now be observed and aggregated on multiple scales, and observed over time. The ECC helps to ensure the completeness of metadata based on the EML standard, and overall, is helping to reduce the cost of curating consistent, tractable datasets.

Funding

This work was supported by the National Science Foundation [grant numbers OCE-0620276, OCE-1232779, and Cooperative Agreements DEB-0832652 and DEB-0936498].

Acknowledgements

We are grateful to the participants on the 2012 workshop and members of the LTER Information Management working group on quality assessment: S. Bohm, E. Boose, J. Downing, M. Gastil-Buhl, C. Gries, B. Leinfelder.

References

American Association for the Advancement of Science, 2011. Dealing with data. Spec. Sect. Sci. 311, 692–729http://www.sciencemag.org/site/special/data/ (accessed 2016-07-20).
American Geophysical Union, 2013. Data Management Maturity Program. http://dataservices.agu.org/dmm (accessed 2016-04-18).
Beek, W.L.R., Bazoobandi, H.R., Weilemaker, J., Schloback, S., 2014. LOD Laundromat: a uniform way of publishing other people's dirty data. In: Mika, P., Bernstein, A., Welty, C., Knocblock, C., Vrandecic, D., Groth, P., Noy, N., Janowicz, K., Goble, C. (Eds.), The Semantic Web – ISWD 2014, Proceedings, Part 1, pp. 213–228.
Eaton, B.J., Gregory, B., Drach, K., Taylor, S., Hankin, J., Caron, R., Signell, P., Bentley, G., Rappa, H., Höck, A., Pamment, M.J., 2011. NetCDF Climate and Forecast (CF) Metadata Conventions. Version 1.6, 5 December 2011 http://cfconventions.org/cf-conventions/v1.6.0/cf-conventions.html (accessed 2016-04-01).
EML Project, 2008. Ecological Metadata Language. https://knb.ecoinformatics.org/#tools/eml (accessed 2016-04-18).
Fegraus, E.H., Andelman, S., Jones, M.B., Schildhauer, M.P., 2005. Maximizing the value of ecological data with structured metadata: an introduction to ecological metadata language (EML) and principles for metadata creation. Bull. Ecol. Soc. Am. 86, 158–168. http://dx.doi.org/10.1890/0012-9623(2005)86[158:MTVOED]2.0.CO;2.
Gosz, J.R., Waide, R.B., Magnuson, J.J., 2010. Twenty-eight years of the US-LTER program: experience, results and research questions. In: Muller, et al. (Eds.), Long-Term Ecological Research. Springer Science + Business Media B. V. http://dx.doi.org/10.1007/978-90-481-8782-9_5.
Habermann, T., 2014. Metadata Evaluation and Improvement. Figshare. http://dx.doi.org/10.6084/m9.figshare.1133879.v1 (accessed 2014-08-11).
Leinfelder, B., Tao, D., Costa, M., Jones, B., Servilla, M., O'Brien, M., Burt, C., 2008. Using metadata for loading and querying heterogeneous scientific data. Pages 90–95. In: Gries, C., Jones, M.B. (Eds.), Proceedings of Environmental Information Management Conference 2008.
Leinfelder, B.J., Tao, D., Costa, M., Jones, B., Servilla, M., O'Brien, M., Burt, C., 2010. A metadata-driven approach to loading and querying heterogeneous scientific data. Ecol. Inform. 5, 3–8. http://dx.doi.org/10.1016/j.ecoinf.2009.08.006.
Lin, C.-C., Porter, J.H., Hsiao, C.-W., Lu, S.-S., Jeng, M.-R., 2008. Establishing an EML-based data management system for automating analysis of field sensor data. Taiwan J. For. Sci. 23, 279–285.
LTER, 2011. EML Best Practices for LTER Sites. Information Management Committee, Long Term Ecological Research Network. https://im.lternet.edu (accessed 2016-04-18).
LTER. 2013. LTER Controlled Vocabulary. http://vocab.lternet.edu (accessed 2016-07-25)

Michener, W.K., Porter, J., Servilla, M., Vanderbilt, K., 2011. Long term ecological research and information management. Ecological Informatics 6 (1), 13–24. http://dx.doi.org/10.1016/j.ecoinf.2010.11.005.

NOAA, 2014. Enterprise Documentation Metrics. https://geo-ide.noaa.gov/wiki/index.php?title=Enterprise_Documentation_Metrics (accessed 2016-07-20).

Peng, G., Privette, J.L., Kearns, E.J., Ritchey, N.A., Ansari, S., 2015. A unified framework for measuring stewardship practices applied to digital environmental datasets. Data Sci. J. 13, 231–253. http://dx.doi.org/10.2481/dsj.14-049.

Peters, D.P.C., 2010. Accessible ecology: synthesis of the long, deep, and broad. Trends Ecol. Evol. 25 (10), 592–601. http://dx.doi.org/10.1016/j.tree.2010.07.005.

Porter, J.H., 2010. A brief history of data sharing in the U.S. Long Term Ecological Research Network. Bull. Ecol. Soc. Am. 14–20.

Porter, J.H., Hanson, P.C., Lin, C., 2012. Staying afloat in the sensor data deluge. Trends in ecology and evolution, February 2012. Vol. 27, 121–129. http://dx.doi.org/10.1016/j.tree.2011.11.009.

Servilla, M., Brunt, J., McGann, J., Costa, D., Waide, R., 2016. The contribution and reuse of LTER data in the provenance aware synthesis tracking architecture (PASTA) data repository. Ecol. Informatics. 36, 247–258.

Tania, A., Candela, L., Castelli, D., 2013. Dealing with metadata quality: the legacy of digital library efforts. Inf. Process. Manag. 49, 1194–1205. http://dx.doi.org/10.1016/j.ipm.2013.05.003.

Zabala, A., Riverola, I., Serral, P., Diaz, V., Lush, J., Maso, Z., Pons, T., Habermann, 2013. Rubric-Q: Adding quality-related elements to the GEOSS Clearinghouse datasets. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 6, 1939. http://dx.doi.org/10.1109/JSTARS.2013.2259580.

Costa, D., 2016. EML congruence checker reports from the long term ecological research network PASTA system, 2013-2015. Long Term Ecol. Res. Netw. http://dx.doi.org/10.6073/pasta/91f41a7d08a09a1567552b0de4ba686f.

O'Brien, M., Costa, D., Servilla, M., Leinfelder, B., Bohm, S., Downing, J., Gastil-Buhl, M., 2016. Congruence checks for EM-described datasets (2015). Long Term Ecological Research Network http://dx.doi.org/10.6073/pasta/7c742b9a5af9fe8ca1762477ec2aa391.