1 Introduction

1.1 Dryad Project

The Dryad project is a funded National Science Foundation grant under the Directorate for Biological Sciences' Biological Databases and Informatics program. The goal of the Dryad project is to develop a metadata and data provider that supports the discovery and management of data associated with publications in the field of evolutionary biology. In addition, the Dryad repository will act as a clearing-house for data/metadata from related fields in biological and environmental sciences. To achieve data/metadata interoperability, the Dryad will adopt the OAI-PMH standard and become both an OAI-PMH “repository” and “harvester”, thereby exchanging metadata (and references to data) with other OAI-PMH compliant repositories. A sub-goal of the Dryad project is to perform metadata exchange with the Metacat XML database that underlies the LTER Data Catalog. As such, this particular goal is the focus of this document, and requires the design and development of an OAI-PMH service interface for Metacat. The Dryad project (and Metacat) will also support implementation of the Library of Congress Search and Retrieve via URL (SRU) standard, which will allow on-the-fly access to repository contents by third parties through a web-service protocol, and will also enable syndication of repository contents (the SRU work is scheduled for year 2 of the NESCent/Dryad LTER subcontract).

1.2 OAI-PMH

The Open Archives Initiative – Protocol for Metadata Harvesting was first developed in the late 1990's as a standard for harvesting metadata from distributed metadata/data repositories. The current version of the OAI-PMH standard is 2.0 as of June 2002, with minor updates in December 2008 (http://www.openarchives.org/OAI/ openarchivesprotocol.html).

The OAI-PMH standard uses the Hyper-text Transport Protocol (HTTP) as a transport layer and specifies six query methods (called “verbs”) that must be supported by an OAI-PMH compliant “repository” (see Section 7.1). These methods are:

1. **GetRecord** – retrieves zero or one complete metadata record from a repository;
2. **Identify** – retrieves information about a repository;
3. **ListIdentifiers** – retrieves zero or more metadata record “headers” (not the complete metadata record) from a repository;
4. **ListMetadataFormats** – retrieves a list of available metadata record formats supported by a repository;
5. **ListRecords** – retrieves zero or more complete metadata records from a respository; and

6. **ListSets** – retrieves the set structure from a repository.

The OAI-PMH compliant repository must accept requests in both HTTP-GET and HTTP-POST formats. Responses from the repository must be returned as an XML-encoded (version 1.0) stream. Error handling must be supported by the repository and provide the correct error response code back to the harvester. Detailed specifications and examples of all six methods may be viewed in Section 4 of the standards document (http://www.openarchives.org/OAI/openarchivesprotocol.html# ProtocolMessages).

1.3 Metadata Cross-walks

1.3.1 EML and DC

Effort will include identifying the mapping of content elements between the EML and DC. The EML (described above) is a fine-grained structure that has many content elements in common with DC. For transformations from the EML to DC, only "simple" or "unqualified" Dublin Core (using only the 15 canonical elements) will be the target of the transformation – OAI-PMH requires unqualified Dublin Core metadata be supported as a minimum. The cross-walk mapping will be approved by the primary contractor prior to development of the XSLT script. Once the mapping elements are identified, an Extensible Stylesheet Language Transformation (XSLT) script will be written for performing the actual transformation. The XSLT will be tested for accuracy and completeness.

1.3.2 EML and Dryad Application Profile

Effort will include identifying the mapping of content elements between the EML and Dryad Application Profile (DAP). The DAP includes content elements from DC, the Data Documentation Initiative, Darwin Core, EML, and PREMIS. The cross-walk mapping will be approved by the primary contractor prior to development of the XSLT script. Once the mapping elements are identified, an Extensible Stylesheet Language Transformation (XSLT) script will be written for performing the actual transformation. The XSLT will be tested for accuracy and completeness.

1.4 OAI-PMH Metacat Service Interfaces

Effort will include designing and implementing an OAI-PMH Metacat "repository" (also referred to as "data provider") and "harvester" service interfaces.

1.4.1 Repository (Data Provider)

The design and implementation of the OAI-PHM Metacat Repository service interface will make available all six OAI-PMH methods (GetRecord, Identify, ListIdentifiers, ListMetadataFormats, ListRecords, and ListSets) as defined in the OAI-PMH Version 2 Specification (http://www.openarchives.org/OAI/

openarchivesprotocol.html) through a standard HTTP URL that accepts both HTTP-GET and HTTP-POST formats. The design will attempt to be metadata neutral with respect to other metadata standards residing within a Metacat instance, however the implementation will only address EML metadata. The design will utilize example OAI-PMH repositories (e.g., OCLC's OAIcat, UIUC provider, or the DLESE provider) as guidance.

1.4.2 Harvester

The design and implementation of the OAI-PMH Metacat Harvester service interface will utilize all six OAI-PMH methods to request metadata or related information from another OAI-PMH compliant repository using a standard HTTP URL in either an HTTP-GET or HTTP-POST format and transform such metadata into the EML standard, which will subsequently be inserted into a Metacat instance. The design will initially harvest DAP data, allowing fall-back to Dublin Core when DAP is not available. The design will allow harvesting of other metadata types in the future. The design will utilize example OAI-PMH harvesters (e.g., OCLC's OAIHarvester2, UIUC harvester, or the DLESE harvester) as guidance.

2 Development Plan

2.1 Metacat SVN Integration

All project related artifacts, including source code, templates, examples, and/or documentation will be integrated directly with the ecoinformatics.org Metacat project from the project beginning. As such, all artifacts will be checked into and managed by the ecoinformatics.org Subversion version control system server (https://code.ecoinformatics.org) and under the “code/metacat/trunk” project directory.

A directory hierarchy follows:

- for design/planning documents - metacat/docs/dev/oaipmh
- for shell scripts and other resources - metacat/lib/oaipmh
- for Java source code - metacat/src/edu/ucsb/nceas/metacat/oaipmh

2.2 Design Details

2.2.1 Crosswalks

EML to DC crosswalk

The following table summarizes the element mappings of the EML to DC crosswalk, including notes specific to each element mapping.

EML Element	DC Element	Notes
Title	title	One-to-one mapping of content
Creator	creator	Use only the creator's name (givenName and surName elements); could be an organization name
keyword	subject	One subject element per keyword element
abstract	description	Must extract text formatting tags
publisher	publisher	Use only the publisher's name (givenName and surName elements); could be an organization name
associatedParty	contributor	Use only the party's name (givenName and surName); could be an organization name
pubDate	date	One-to-one mapping
dataset | citation | protocol | software	type	Type value is determined by the type of EML document rather than by a specific field value
physical	format	Use a mime type as the Format value? For example, if EML has <textFormat> element within <physical>, then use 'text/plain' as the Format value?
(1) packageId; (2) URL to the EML document	identifier	packageId can be used as the value of one identifier element; a second identifier element can hold a URL to the EML document
dataSource	source	Use the document URL of the referenced data source?
Citation	relation	Use the document URL of the referenced citation?

geographicCoverage	coverage	Add separate coverage elements for geographic description and geographic bounding coordinates.
		For bounding coordinates, use minimal labeling, for example:
		`81.505000 W, 81.495000 W,`
		`31.170000 N, 31.163000 N`

taxonomicCoverage	coverage	Use only genus/species binomials; place each binomial in a separate coverage element
temporalCoverage	coverage	Include begin date and end date when available. For example: `1915-01-01 to 2004-12-31`
intellectualRights	rights	Must extract text formatting tags

2.2.2 Repository (Data Provider)

The Metacat OAI-PMH Repository service was implemented using the Online Computer Library Center (OCLC) OAICat Open Source Software as the basis for the implementation, with substantial customizations and modifications added to facilitate integration with Metacat.

Customizations and additions to the set of OAICat classes are described in the following table:

Package Name	Class Name(s)	Modified Class vs. New Class	Description
edu.ucsb.nceas.metacat.oaipmh.provider.server	OAIHandler.java	Modified	OAIHandler is the primary servlet class for OAICat. It has been customized to allow loading of Metacat configuration values.
edu.ucsb.nceas.metacat.oaipmh.provider.server.catalog	MetacatCatalog.java	New	MetacatCatalog is an implementation of AbstractCatalog interface. It is responsible for determining which documents exist in Metacat, their document types, and their datestamps.
edu.ucsb.nceas.metacat.oaipmh.provider.server.catalog	MetacatRecordFactory.java	New	MetacatRecordFactory, a subclass of RecordFactory, converts native Metacat documents to OAI-PMH records.
edu.ucsb.nceas.metacat.oaipmh.provider.server.crosswalk	Eml200.java Eml201.java Eml210.java	New	The set of Eml2xx.java classes are responsible for retrieving EML documents in their native format. Although these classes are subclasses of the Crosswalk class, no actual XSLT transformation is performed by them.
edu.ucsb.nceas.metacat.oaipmh.provider.server.crosswalk	Eml2oai_dc.java	New	Eml2oai_dc, a subclass of Crosswalk, is responsible for retrieving EML 2.x.y documents and transforming them to oai_dc (Dublin Core) format.

With the exception of OAIHandler.java, all other native OAICat classes are used in their original (non-modified) form and are contained in the Java archive file:

metacat/lib/oaipmh/oaicat.jar

The XSLT stylesheets used by the Eml2oai_dc.java class reside in files:

metacat/lib/oaipmh/eml200toDublinCore.xsl

metacat/lib/oaipmh/eml201toDublinCore.xsl

metacat/lib/oaipmh/eml210toDublinCore.xsl

Repository Design Issues

1. 'Deleted' Status

OAI-PMH repositories can optionally flag records with a 'deleted' status, indicating that a record in the metadata format specified by the `metadataPrefix` is no longer available. Since the Metacat database does not provide a readily apparent mechanism for retrieving a list of deleted documents, the use of the 'deleted' status is not supported in this implementation of the OAI-PMH repository. This represents a possible future enhancement to the repository design.

2. Sets

OAI-PMH repositories can optionally support set hierarchies. Since it has not been determined how set hierarchies should be structured in Metacat, this implementation of the OAI-PMH repository does not support set hierarchies. This represents a possible future enhancement to the repository design.

3. Datestamp Granularity

When expressing datestamps for repository documents, OAI-PMH allows two levels of granularity – ***day granularity*** and ***seconds granularity***. Since the Metacat database stores the value of its `xml_documents.date_updated` field in day granularity, that is the level that is supported by the Metacat OAI-PMH Repository.

2.2.3 Harvester

The Metacat OAI-PMH Harvester client was implemented using OCLC's OAIHarvester2 open source code as its base implementation, with customizations and additions as needed to support integration with Metacat.

Customizations and additions to the set of OAIHarvester2 classes are described in the following table:

Package Name	Class Name(s)	Modified Class vs. New Class	Description
edu.ucsb.nceas.metacat.oaipmh.harvester	OaipmhHarvester.java	New	OaipmhHarvester is the primary Java class for executing the code, containing the main() method. It is a heavily modified version of the RawWrite.java class included in the OAIHarvester2 source code.
edu.ucsb.nceas.metacat.oaipmh.harvester	HarvesterVerb.java	Modified	HarvesterVerb is a parent claass to the six verb subclasses. This is a heavily modified version of the original HarvesterVerb class in the OAIHarvester2 source code.
edu.ucsb.nceas.metacat.oaipmh.harvester	GetRecord.java ListIdentifiers.java	Modified	These two verb classes are used to drive the harvest engine. The program first runs the 'ListIdentifiers' verb to determine which records exist in the remote respository, as well as their datestamps. Then it runs the 'Get Record' verb, as needed, to retrieve individual records from the remote respository for insertion/update into Metacat.
edu.ucsb.nceas.metacat.oaipmh.harvester	Identify.java ListRecords.java ListSets.java ListMetadataFormats.java	Unmodified	These verb classes are essentially unmodified from their corresponding classes in the OAIHarvester2 source code. They are not actively used by the Metacat OAI-PMH Harvester code,

			but are included with the source code for completeness and for potential future development.

Harvester Design Issues

1. Dryad Identifiers

Dryad stores documents using identifiers like the following:

oai:dryad-dev.nescent.org:10255/dryad.12

When harvested into Metacat, these identifiers are converted to a Metacat-legal equivalent, for example:

10255-dryad.12

(Note the use of a '-' character in place of a '/' character. It was discovered during development that the '/' character caused errors when used in Metacat identifiers.)

2. Handling of 'Deleted' status

The Metacat OAI-PMH Harvester program *does* check to see whether a 'deleted' status is flagged for a harvested document, and if it is, the document is correspondingly deleted from the Metacat repository.

3. Datestamp Granularity

When expressing datestamps for repository documents, OAI-PMH allows two levels of granularity – ***day granularity*** and ***seconds granularity***. Since the Metacat database stores the value of its 'xml_documents.last_updated' field in day granularity, that is the level that is supported by both the Metacat OAI-PMH Repository and the Metacat OAI-PMH Harvester. This has implications when Metacat OAI-PMH Harvester (MOH) interacts with the Dryad repository, which stores its documents with seconds granularity. For example, consider the following sequence of events:

1. On January 1, 2010, MOH harvests a document from the Dryad repository with datestamp '2010-01-01T10:00:00Z', and stores its local copy with datestamp '2010-01-01'.
2. Later that same day, the Dryad repository updates the document to a newer revision, with a new datestamp such as '2010-01-01T20:00:0Z'.
3. On the following day, MOH runs another harvest. It determines that it has a local copy of the document with datestamp '2010-01-01' and *does not* re-harvest the document, despite the fact that its local copy is not the

latest revision.

3 Configuring and Running

3.1 *Configuring and Running the Metacat OAI-PMH Data Provider Servlet*

1. Uncomment the **`servlet-name`** and **`servlet-mapping`** entries for the **`DataProvider`** servlet in file:

   `metacat/lib/web.xml.tomcat5`

2. Edit properties in the OAI-PMH section of file

   `metacat/lib/metacat.properties`

The following table describes the properties stored in `metacat.properties` that are used by the DataProvider servlet:

Property Name	**Sample Value**	**Description**
oaipmh.maxListSize	5	Maximum number of records returned by each call to the ListIdentifiers and ListRecords verbs.
oaipmh.repositoryIdentifier	metacat.lternet.edu	An identifier string for the respository.
AbstractCatalog. oaiCatalogClassName	edu.ucsb.nceas.metacat.oaipmh. provider.server.catalog. MetacatCatalog	The class that implements the AbstractCatalog interface. This class determines which records exist in the repository and their datestamps.
AbstractCatalog. recordFactoryClassName	edu.ucsb.nceas.metacat.oaipmh. provider.server.catalog. MetacatRecordFactory	The class that extends the RecordFactory class. This class creates OAI-PMH metadata records.
AbstractCatalog. secondsToLive	3600	The lifetime, in seconds, of the resumptionToken.
AbstractCatalog. granularity	YYYY-MM-DD *or* YYYY-MM-DDThh:mm:ssZ	Granularity of datestamps. Either 'days granularity' or 'seconds granularity' values can be used.
Identify.repositoryName	Metacat OAI-PMH Data Provider	A name for the repository.
Identify.earliestDatestamp	2000-01-01T00:00:00Z	Earliest datestamp supported by this repository
Identify.deletedRecord	yes *or* no	Use 'yes' if the repository indicates the status of deleted records; use 'no' if it doesn't.
Identify.adminEmail	mailto:tech_support@LTERnet.edu	Email address of the repository administrator.
Crosswalks.oai_dc	edu.ucsb.nceas.metacat.oaipmh. provider.server.crosswalk. Eml2oai_dc	Class that controls the EML to oai_dc crosswalk.
Crosswalks.eml2.0.0	edu.ucsb.nceas.metacat.oaipmh. provider.server.crosswalk.	Class that furnishes EML 2.0.0 metadata.

	Eml200	
Crosswalks.eml2.0.1	edu.ucsb.nceas.metacat.oaipmh.provider.server.crosswalk.Eml201	Class that furnishes EML 2.0.1 metadata.
Crosswalks.eml2.1.0	edu.ucsb.nceas.metacat.oaipmh.provider.server.crosswalk.Eml210	Class that furnishes EML 2.1.0 metadata.

The properties whose values will need to be changed for a particular Metacat installation are the `oaipmh.repositoryIdentifier` and the `Identify.adminEmail` properties. The remaining properties can usually be left as is, retaining their default values as they appear in the `metacat.properties` file.

Example URLs

Examples of URLs that demonstrate use of the Data Provider servlet follow:

OAI-PMH Verb	**Description**	**URL**
GetRecord	Get an EML 2.0.1 record using its LSID identifier	http://scoria.lternet.edu:8080/knb/dataProvider?verb=GetRecord&metadataPrefix=eml-2.0.1&identifier=urn:lsid:knb.ecoinformatics.org:knb-lter-gce:26
GetRecord	Get an oai_dc record using its LSID identifier	http://scoria.lternet.edu:8080/knb/dataProvider?verb=GetRecord&metadataPrefix=oai_dc&identifier=urn:lsid:knb.ecoinformatics.org:knb-lter-gce:26
Identify	Identify this data provider	http://scoria.lternet.edu:8080/knb/dataProvider?verb=Identify
ListIdentifiers	List all EML 2.1.0 identifiers in the repository	http://scoria.lternet.edu:8080/knb/dataProvider?verb=ListIdentifiers&metadataPrefix=eml-2.1.0
ListIdentifiers	List all oai_dc identifiers in the repository between a range of dates	http://scoria.lternet.edu:8080/knb/dataProvider?verb=ListIdentifiers&metadataPrefix=oai_dc&from=2006-01-01&until=2010-01-01
ListMetadataFormats	List metadata formats supported by this repository	http://scoria.lternet.edu:8080/knb/dataProvider?verb=ListMetadataFormats
ListRecords	List all EML 2.0.0 records in the repository	http://scoria.lternet.edu:8080/knb/dataProvider?verb=ListRecords&metadataPrefix=eml-2.0.0
ListRecords	List all oai_dc records in the repository	http://scoria.lternet.edu:8080/knb/dataProvider?verb=ListRecords&metadataPrefix=oai_dc

ListSets	List sets supported by this repository	http://scoria.lternet.edu:8080/knb/dataProvider?verb=ListSets

3.2 Running the Metacat OAI-PMH Harvester

The Metacat OAI-PMH Harvester (MOH) is executed as a command-line program:

```
sh runHarvester.sh -dn <distinguishedName>  \
                   -password <password>  \
                   -metadataPrefix <prefix>  \
                   [-from <fromDate>] \
                   [-until <untilDate>] \
                   [-setSpec <setName>] \
                   <baseURL>
```

The following example illustrates how the MOH is run from the command line:

```
% cd $METACAT_HOME/lib/oaipmh
% sh runHarvester.sh -dn uid=dryad,o=LTER,dc=ecoinformatics,dc=org  \
                     -password some_password  \
                     -metadataPrefix oai_dc  \
                     http://baseurl.repository.org/knb/dataProvider
```

Command line options and parameters are described in the following table:

Command Option or Parameter	**Example**	**Description**
-dn	-dn uid=dryad,o=LTER,dc=ecoinformatics,dc=org	Full distinguished name of the LDAP account used when harvesting documents into Metacat. (Required)
-password	-password some_password	Password of the LDAP account used when harvesting documents into Metacat. (Required)
-metadataPrefix	-metadataPrefix oai_dc	The type of documents being harvested from the remote repository. (Required)
-from	-from 2000-01-01	The lower limit of the datestamp for harvested documents.

		(Optional)
-until	-until 2010-12-31	The upper limit of the datestamp for harvested documents. (Optional)
-setSpec	-setSpec someSet	Harvest documents belonging to this set. (Optional)
base_url	http://baseurl.repository.org/knb/dataProvider	Base URL of the remote repository

4 Generalized Definitions

4.1 OAI-PMH

(*see OAI-PMH specification Section 2 for details - http://www.openarchives.org/OAI/ openarchivesprotocol.html# DefinitionsConcepts)*

- **Datestamp** – a datestamp is an optional construct for categorizing or grouping items based on a time frame for the purpose of “selective harvesting”; datestamps use a “day” granularity and can specify either or both beginning and ending values; datestamps can be used to identify creation, modified, and deletion events associated with an “item”.
- **Harvester** – a client application that use the OAI-PMH methods to retrieve metadata from and/or information about a “repository”.
- **Item** – an informational container of metadata within a “repository” that can serve as the origin of content when constructing an OAI-PMH metadata record; each “item” is considered unique within the “repository”.
- **Record** – metadata expressed as a single format and as an XML-encoded stream that is returned in response to an OAI-PMH request; each record consists of a “header”, “metadata”, and “about” (optional) sections. The concept of “deleted records” must be supported by the “repository”.
- **Repository** – a “network accessible server” that hosts metadata and complies with the OAI-PMH standard for serving metadata requests.
- **Selective Harvesting** – selective harvesting allows a “harvester” to limit harvest requests to subsets of the metadata available from a “repository” based on either “datestamps” and/or “sets”.
- **Set** – a set is an optional construct for categorizing or grouping items based on a logical theme for the purpose of “selective harvesting”; a set can be flat or hierarchical.
- **Unique Identifier** – a unique identifier (UID) unambiguously identifies an “item” in a repository; the UID must conform to the Uniform Resource Identifier (IETF RCC 2396) syntax (http://www.ietf.org/rfc/rfc2396.txt?number=2396).

5 Informational Links

- **Dublin Core Metadata Initiative** – http://dublincore.org/.
- **Ecological Metadata Language** (EML) – http://knb.ecoinformatics.org/software/eml
- **Knowledge Network for Biocomplexity** (KNB) (NSF DEB99–80154) – http://knb.ecoinformatics.org.
- **LTER Data Catalog** – http://metacat.lternet.edu
- **Metacat** – http://knb.ecoinformatics.org/software/metacat/.
- **OCLC OAICat** - http://www.oclc.org/research/software/oai/cat.htm

  OAICat was used as the base implementation of the Metacat OAI-PMH Repository (the DataProvider servlet).
- **OCLC OAIHarvester2** - http://www.oclc.org/research/software/oai/harvester2.htm

  OAIHarvester2 was used as the base implementation of the Metacat OAI-PMH Harvester.
- **Open Archives Initiative** – http://www.openarchives.org/

6 OAI-PMH Error Codes

Error Code	***Description***	***Applicable Verbs***
badArgument	The request includes illegal arguments, is missing required arguments, includes a repeated argument, or values for arguments have an illegal syntax.	*all verbs*
badResumptionToken	The value of the resumptionToken argument is invalid or expired.	ListIdentifiers ListRecords ListSets
badVerb	Value of the verb argument is not a legal OAI-PMH verb, the verb argument is missing, or the verb argument is repeated.	N/A
cannotDisseminateFormat	The metadata format identified by the value given for the metadataPrefix argument is not supported by the item or by the repository.	GetRecord ListIdentifiers ListRecords
idDoesNotExist	The value of the identifier argument is unknown or illegal in this repository.	GetRecord ListMetadataFormats
noRecordsMatch	The combination of the values of the from, until, set and metadataPrefix arguments results in an empty list.	ListIdentifiers ListRecords
noMetadataFormats	There are no metadata formats available for the specified item.	ListMetadataFormats
noSetHierarchy	The repository does not support sets.	ListSets ListIdentifiers ListRecords