Raw Notes From Etherpad for July 2011 CCIT Meeting, Santa Barbara

Source: http://epad.dataone.org/20110719-CCIT

Revision: Etherpad 9

Contents

About

These are the raw notes captured in Etherpad. Content has been slightly modified for rendering / spelling.

CCIT Face to Face Meeting Notes

Date: 2011-07-19 to 2011-07-21

Location: NCEAS, Santa Barbara

Attendees: Amber B., Roger D., Mark S., Dave V., Ryan S., John K., Jeff H., Bob S., Giri P., Chris J., Bruce W., Paul A., Robert W., Matt J., Ben L., Ryan K., Nick D.

July 19, Morning Session

See Dave's presentation (https://repository.dataone.org/documents/Committees/CCIT/20110719_CCIT_SantaBarbara/docs/20110719_D1_CI_notes.pptx)

Goals for 2011

- public release of cyberinfrastructure
- coordinating nodes
- 3 CNs
- member nodes
- need min. of 6 MNs up and running

Dryad

The most challenging part of getting Dryad and KNB to tier 2 due to mapping the internal user identify to D1 identities.

Dryad is planning to support replication (tier 4) but Dryad has institutional restrictions on accepting data directly (tier 3) since data must be linked to publications. Providing tier 3 functionality doesn't mean the MN has to accept data from the world, but could accept writes only from its local community using ITK tools.

CUAHSI

Has challenges with streaming data; DataONE interfaces (pre-refactoring to tiers) are mapped to CUAHSI services, but are lacking checksums and file sizes due to streaming nature of data

D1 will hopefully deal with streaming data next year; we should deal with chunked data now and streamed data later

Also need to have a mechanism to package chunks of streams and then a way for end users to compile chunks into a whole - this is also a discovery issue (e.g., how to find all of the chunks that make up a whole dataset)

KNB

Matt: KNB currently has usability issues due to individual chunks of data deposited, but no consistent way to identify all the chunks belonging to a whole

Currently debating between pushing MN implementaiton to HIS central and individual hydro servers. HIS Central has a metadata catalog that knows about the contents of all of the individual HydroServers. If we had a Member Node at HIS Central it could give us access to a lot more data.

Individual hyrdro servers providing access control, and HIS central could be setup as a MN but would still have to delegate access control to individual hyrdo servers.

AKN

- D1 intern was supposed to tackle AKN MN implemented in Metacat; but didn't work out
- haven't identified resources to implement
- Not seen as a huge effort, but still some dedicated time needs to be found. No local authentication/authorization mechanism (same as CUAHSI). Would have just public-readable data (which is most of AKN -- just a few datasets that aren't public). ??? how run metacat member node with this public readable. Ben has been working on a new version that's not LDAP dependant -- would use CI Logon and InCommon.
- Could transition to accepting data deposits via Metacat instead of traditional AKN ingest mechanisms

Merritt

- Expect to be a short distance supporting tier 1.
- Authentication is HTTP Basic over SSL
- John: Nothing yet implemented, but still feels realistic to get node up by end of year.

Fedora

- not a candidate for this year

Replication Targets

- implemented with GMN
- needs to be brushed off and tested; also stress tested
- 1000 objects tested; need to test and order of magnitude or two more

TeraGrid MN

- based on Metacat
- not ready to commit to having this TeraGrid MN hooked up to production CNs; the hope is to have it hooked up to the staging CN infrastructure but we aren't ready to commit to supporting it in the production CN infrastructure

Mercury-based MN

- ready for tier 1 testing

- will support ORNL DAAC, and NBII
- can support both both local data dn data not directly held
- had plan to release NBII MN to D1 by end of year (will be ready for alpha testing by the end of August)
- Dave: What about overlapping content in NBII? Giri: Can pretty easily filter out duplicate metadata.
- Using DOIs/EZIDs as primary identifiers
- Will need to also filter out data replications form KNB.

Coordinating Nodes (0.6.2 API version)

- Object storage is implemented
- Identifier resolution is implemented (resolve())
- Synchronization
  - Robert using Hazelcast (http://www.hazelcast.com/) for synchronization. Immediately consistent model (rather than eventually consistent). Not clear how this handles the network partitioning case.
  - Reference back to paper Matt Jones saw a few years ago which asserts (one data point) that 36 distributed nodes across the internet are needed to get an assessment of network partitioning and location.
  - Need to filter out existent sciMeta objects via search such that replication activities does not increase load on CN.

Logging

- Log summaries would be generated on a daily basis for a given object

Identity Mapping

- Should we use an internal D1 identity and map it to the many identities that are equivalent, including the CILogon identity?

Session State

- Issues: session maintenance isn't handled by CILogon

Identifier reservation

- API is in place
- Back end store is in place
- Identifiers are in LDAP
- hasReservation() is in 0.6.2
- reservation lasts 24 hours
- Issues: reservation of a pid is dependent on a call to a CN, which requires network connectivity. The subsequent create() call is also dependent on the CN

Replication

- replicate() is implemented on the Metacat-based MN and the GMN, but isn't tested because the CN component is not ready (coordinating the target replication MN)
- Roger's suggestion: an MN should be able to provide Tier 4 MNReplication without implementing tiers 1,2, and 3. This is desirable, but may not be practical from a prioritization standpoint
- CN should check the system metadata of an object to determine replica status before attempting an expensive replicate call to an MN that may already have a replica

Testing

- Rob is working on a web testing framework/dashboard. Can we run the KNB Metacat implementation against this to test the MN implementation? This is useful for bringing up new MNs to allow for full API testing before adding the MN to the node list
- Need to refactor d1_integration to gain access to src/test/java classes (the actual integration tests) from the deployed servlet context
- Need to address authentication issues in the client for nodes requiring https: / SSL
- Pre-conditions for new MNs to use:
  - some objects to find to test Tier1 "gets"
  - configuration of the MN to allow access by TBD testing credentials
- Can test methods that do not require interaction with other nodes - envision graduating to a staging environment for testing more complicated use cases (replication, synchronization) when these first tests pass.

ITK

- requirements are more flexible
- see DUG priority list: https://repository.dataone.org/documents/Meetings/20110711_DUG_SantaFe/itk-tool-prioritization-by-lifecycle.pdf
- R plugin - not updated since 0.5.1 (Feb 2011)
- D1Drive - FUSE not updated since 0.5.1 (Feb 2011)
- D1Drive - Dokan - functionality is similar to FUSE. Limited to testing without an object list being returned by the CN
  - The D1 user group felt that the D1Drive needs to be launchable via a graphical application vs the command line
- Mercury UI needs to support search with access control (need to inject Session Subject via the SSL certificate) - Addressed most of the usability WG suggestions - adding enhanced map selection to support polygon searches - adding thumbnails in the search results to display the spatial coverage of the dataset - solr index enhancements - distributed granule search using openSearch (currently implemented in ORNLDAAC, but this can be used for D1 in the future) - search enhancement/optimization - need to update the stylesheet to match with the latest D1 look and feel - Note that the D1Drive calls the Solr API, and it's effectively not documented in D1 documentation

Other Pieces

Documentation

MN test service (web testing)

Integration testing

Monitoring - currently using Cacti

Hardware - needs purchasing and deployment

- UCSB has North Hall location, hardware is ordered?, virtualization is KVM?
- Shooting for mid-August installation date
- ORNL has the 'Spam Can' location, hardware is ordered, virtualization is VMWare
- Looking at end of July delivery
- UNM leveraging hardware in place, and purchasing hardware to augment, location is the UNM Research Library, virtualization will be VMWare

July 19- Afternoon Session

Authentication and Authorization

Ben's update:

- Client Certificates are used for:
    - client to MN
    - client to CN
    - used for all service calls, some optional
    - CN certificates
    - CN to MN replication
    - CN to CN synchronization
    - MN certificates
    - MN to MN replication
    - MN to CN node, auth, and identity calls

CILogon specialized skin, log in, get an 18 hr cert

Note that the Google cert has no meaningful attributes included, just a random string, which means each D1 new user looking to use a service will have to register in D1 and map their identity to the Google identity

If D1 has to store credentials, we have basically replicated the authn infrastructure

Multiple Google accounts possible

Bruce: logged in with my refbruce@gmail.com identity in cilogon. Response returned was:

```
/DC=org/DC=cilogon/C=US/O=Google/CN=Bruce Wilson A609
```

Logged in with wilsonbe@ornl.gov Google Apps identity. Response was:

```
/DC=org/DC=cilogon/C=US/O=Google/CN=Bruce Wilson A607
```

Interesting: Chrome login with ornl.gov identity just allowed downloading the certificate. Firefox with gmail.com identity prompted for a password to protect the cert. -- Found the problem. Difference in URL for the login.

Does Google Two-factor login work? Mark: yes.

It may be problematic to map identities because we can't easily gain a comprehensive Subject list to the D1 service to allow manual mapping

??? Will this work with CILogon for things that don't really have a file system (like iOS and Androd)? Probably won't work on an iPad or a ChromeBook?

In restricting access to services (full collections, not just resource objects), we need to be able to express an authorized subject list

This can be done in the node list, or in an access policy - decision:

CILogon benefits:

- re-use institutional identities
- levels of identity verification

??? What is the requirement for identity verification. Some data providers do want to require that only people with some level of verified accounts.

CILogon and Lynx:

- must accept several cookies
- after logging on at the IDP, it leaves the DataONE skin, and goes to the "create a certificate" screen
- successfully downloads the jnlp
- lynx https://cilogon.org/?skin=DataONE
- SSL error:unable to get local issuer certificate-Continue? (y)
- cilogon.org cookie: PHPSESSID=e2qpd1a1cd2hht3qjvg9servr7 Allow? (Y/N/Always/neVer)
- etc ...

Discussion abut Authentication with CILogon.

General consensus that the CILogon that currently exists is somewhat user hostile, particularly for browser-based access to information. Multiple options exist:

A. (roll our own) DataONE uses it's own authentication provider behind a REST login service, generates its own certificates. The challenge here is that we've predicated a lot of the tools using certificates. We have to become our own certificate provider for this to work. (2)

   - No real idea who people are or real identity.
   - If we put up our own identity provider, we probably need to commit to support that over the long haul (multiple years).

B. Suck it up and use CILogon as it currently stands. (0)

C. Work with CILogon to improve portal service and add app widget (11) Modify Mercury (as the primary web-based toolset) to call a portal for handling the back-end CILogon information. Three pages. Mercury communicates with this portal to get the certificate (via session cookie?). Can CILogon provide this portal (maybe)?

   - Modify the applications to be able to get the certificate inside the application itself.
   - Similar issues with DataONE provider, in that someone can deposit now from an institutional account, and then not come back later.
   - Concern about whether this is achievalble. If miss targets, back in (B), and there is concern about poor initial user experience.

D. Use the back-end InCommon for authentication (Shibboleth) in the web browser applications and use CILogon for the authentication for command line tools. (0)

   - Modifiy the applications to use the Shibboleth InCommon providers. Similar to the same sorts of things that using the InCommon certs do.

E. Traditional non-cert, non shib login. Session/cookie based.

Conclusion Build the portal thing that manages the process of the user connecting to CILogon. Portal requests the certificate, stored on the server side, with an option to download that to the desktop as an option for running desktop applciations. Continue to develop the necessary web browser interfaces to interact with CILogon. Desktop app widget development secondary but desirable, provides ability to login via CILogon within app and obtain certificate within app. Be aware that the private key should not be downloadable from the intermediate service, because we don't want to expose the private key with the potential there for information leakage. Backup plan is that we're in the (E) camp.

Data Packaging

Outstanding questions

1. How to put non-URIs in ORE statements

   - Possibly create a CN.get() that automatically does a resolve and retireves the data to cretae a real URI at that REST endpoint
   - https://cn.dataone.org/cn/object/foo.1.1
   - use a surrogate ID from EZID?

2. Should we have a getPackage() REST service now?

   Not now -- delay until later, but yes later after

3. Do BagIt bags contain all, some, or none of the data?

   Defer until getPackage is resolved

4. Should we have a 'getORE()' call to get just the Map?

   Yes. Create service to retrieve ORE document from any package component

   ```
   getPackageDescription(pid)::Identifier[]
   ```

   - Should be list objects by filtering on pid and objectFormat, but this is a different pid (need to index the pid/pid mapping for objects and their associated ORE)

5. How do we resolve redundancy between SystemMetadata and ORE map (and BagIt manifest)?

   See refactoring proposal below.

6. How are packages represented in listObjects?

   Just returned like everything else, with its own ObjectFormat and identifier.

7. Can packages contain components that themselves lack unique D1 identifiers? **No.**

   For now, take the idea of streaming data and large RS data off the table; revisit later in yr 3.

   Proposal: for finite, static data sets, each object gets its own ID, and is referenced in an ORE map

8. What ORE predicates should be standardly used in our ORE docs?

   desribes, describedBy, derivedFrom; add properties for companionDocuments (can companion documents be described by dcterms:hasPart?)

9. Do we require that an ORE document exist for all data sets?

   No.

10. Should ORE maps only point to DataONE objects, or can they also point at external resources?

    Only D1 resources, at least initially.

    If we start by restricting to only DataONE content, we can always relax later, but not vice versa

    ORNL DAAC example dataset with companion files: http://daac.ornl.gov/cgi-bin/dsviewer.pl?ds_id=14

TODO

- create full ORE example with associated package components
- create Bagit serialization example
- Add ORE to ObjectFormat list
- discuss additional properties in addition to describes for less structured information

Proposal

Keep SystemMetadata for each object, but factor out describes/describedBy and derivedFrom and put that in an ORE map. ORE maps are versioned and have their own identifiers. SystemMetadata gets a new field with a pointer to the list of OREMaps in which it participates (possibly not needed because this is the reciprical relationship from what is in the ORE Map itself). New function to get list of ORE maps for a resource, returns list of Identifiers of associated ORE maps.

ReserveIdentifier:

- two separate methods
  - generateIdentifier(scheme, [fragment]) returns DOI ARK

    ::

        scheme = type of identifier "DOI" or "ARK"

  - reserveIdentifier(string) returns string or fail

July 20, 2011 Discussions

Versioning of Infrastructure

- messages between services may change
- message semantics may change
- behavior of components may change

Infrastructure version will be tagged based on overall functionality

Changes

- change the type schema
- update test data
- build lib_common_* via Pyxb and jibx
- Propogate change through lib_client_*
- Propogate changes to other components using the libs
- Deploy CN test
- Update test MNs
- Deploy staging
- Update/ sync content to match production

Questions:

- where to update MN stacks?
- Where to update ITK?

Discussion

May need to support versions for a year or two to allow MNs to deprecate and move forward

Copying/moving data along with software changes is very difficult

Should the staging move in sync with production?

Downtime may be necessary to migrate CNs

Latency and aceptable deltas

Rolling window scenario where multiple versions are supported and overlap over time

Type definitions: may want to break types out into separate XSDs such that types may change more modularly

LDAP and PostgreSQL data schemas should only have extensions. Don't drop attributes.

Changing the meaning of an attribute should be done by just adding a new attribute.

How can we ensure that the running member nodes can continue to run, allowing the staged movement of CNs and MNs to move forward to the new versions? Assume that there is a staging enviornment, where the new version of the CN is moved out and the staging then gets flipped to be the production enviornment. Consider a case of significant changes to system metadata. How would that process forward? Likely scenario that we may have to be supporting 1.0 and 2.0 clients at the same time (months). Might be able to take 1.0 out of production when we release 3.0.

If new field is added in system metadata, would have to add the field in the database, then modify the parsers.

Discussion

Should store only the latest version of a given type. The version 1 of a call has to be able to read the new version of the object and translate to the old version (get operation from a client). For a client put, the v1 method has to accept the old version and transform to the new version to write this to the database. Need to ensure that the serialization of a given type includes the version of that type. This is common for most XML serializations, in the schema definition version.

??? Can the member node stacks be auto-updated, in the same way that Chrome autoupdates? Could be done for client tools, as a maintenance method. For services like member nodes, this could be a problem.

??? Does this this change if we look at changes in a more incremental, continuous beta fashion, rather than sets of relatively rare changes.

Issue: how does the client signal which version of a method that it's calling? Is this in the URL or is this in the arguments that it sends to that REST URL. One way is to change the REST signature when there is a different signature. The other option is that it's in the header, as a required attribute, and the dispatcher has to figure out what version is in the header and call the appropriate method. Could put it in the message request header, but that does make it difficult to use a web browser because it wouldn't put that into the message header. Could also make it more difficult for the dispatcher, as the processor has to actually read part of the message to determine where the message should go. Putting it in the URL also makes a simple grep to search for what's happening.

Question: Do all versions of the methods move together? Would we put the version number in all of the method signatures? If we did a service pack change and created a new method, for example, would we then change the signature of all other method signature. CLO does the versioning at the REST URL level. Changes happen infrequently. Can add new stuff to an existing level, but semantic changes are a major release and all method signatures update.

If we add a method to a tier, then either that tier has to be versioned, or we have to add a new tier to the system. If we add, for example, a getpackage method and it becomes allowed in Tier 1, is this now a new tier, or are we creating a version of that tier. Similar issue with things like data subsetting operations.

Paul: perspective from the experience of CLO is that making it easy for people to tell what version they're on and what needs to be done to move to the next version. Helps with engagement in groups and simplifying the member node developer's jobs.

Rob: Do XSD schemas support optional elements, such as <version>? It may allow us to avoid schema validation problems and the need for the Java code base to support multiple schema versions and Jibx generated datatype classes. see: "W3C XML Schema Design Patterns: Dealing With Change" - http://msdn.microsoft.com/en-us/library/aa468563.aspx

MN version signalling:

A. URL encoding of version:

```
/mn/meta/{pid}
/mn/0.6.2/meta/{pid}
/mn/v1/meta/{pid} -- Similar to CLO method, all versioned together
/mn/meta/v1/{pid} -- could conflict with {pid}.  Makes it more confusing to read.
/mn/meta/{pid}?v=1
```

B. Node registry information

C. HTTP request headers - version information must appear in request / response
Problem - client must send version info info request

D. Message namespaces

    - requires reading messages to figure out what it is

Versions in URLs indicate consistency in the interface definitions. Any change to interface requries an update to the version tag in the URL. Version tag in URL sould be simple (i.e. not tied to software version) like "v1". Proposed granularity of the Interface version is only major numbers, v1, v2, v3, v4, etc.

from Dave, the dynamics:

```
change in API       ==== requires ====>   change in code  (client/cn/mn impl, integration tests)

change in schema    ==== requires ====>   change in code

change in schema    ==== requires ====>   change in API
```

YET, a change in API does not require change in datatypes

so, datatypes within d1_common_(java) correspond to schema version

For example:

```
starting at:
    api version: v3
    schema version: 0.6.1
    code release: x.0.0
```

1. release of D1_SCHEMA_0_6_2 results in new API version and code version, and new datatypes:

   ```
   api version: v4 for example
   new code version: x.1.0
   and new package in d1_common_java:  org.dataone.service.types.0_6_2
   ```

2. **Further code implementation to existing schemas and service API**

   new code version x.1.1 (is x.2.0 possible in this case?)

3. Following that: release of D1_SCHEMA_0_6_3 triggers:

   ```
   new api version: v5
   new code version x.2.0
   and new package in d1_common_java: org.dataone.service.types.0_6_3
   ```

4. Following that API update:

   ```
   new api version: v6
   new code version x.3.0
   ```

- because api versions will represent the "dataone" version, we should consider using "major.minor" format or else we will be at a high version number very quickly.

Example d1_common structure for the v1 release of d1_common_1.0.0:

```
org.dataone.service.cn.v1.CNAuthorization
org.dataone.mn.tier1.v1.MNCore
org.dataone.mn.tier1.v1.MNRead
org.dataone.mn.tier2.v1.MNAuthorization
org.dataone.service.types.v1.SystemMetadata
```

Example d1_common structure for the v2 release of d1_common_2.0.0:

```
org.dataone.service.cn.v1.CNAuthorization
org.dataone.mn.tier1.v1.MNCore
org.dataone.mn.tier1.v1.MNRead
org.dataone.mn.tier2.v1.MNAuthorization
org.dataone.service.types.v1.SystemMetadata
org.dataone.mn.tier1.v2.MNCore
org.dataone.mn.tier1.v2.MNRead
org.dataone.mn.tier5.v2.MNSubset
```

The problem with multiple api versions within the d1_common package is that the client will have to implement all of these methods again, and the client will be talking many versions. Is there a need for a client to switch which version of the API it's talking? (CN-as-client, MN-as-client?) or can the user include multiple libclient jars in the pom instead?

Actions:

1. Multiple versions MUST be supported
2. Change REST urls to include infrastructure version
    - versioning at the Tier level. All REST URLs for a tier have the same version tag
3. Base URL of CNs and MNs MUST return node registry information for the node
    - cn.dataone.org -> Human interface
    - cn.dataone.org/cn -> node registry doc
3. Define a process for interface versioning independent of the software tags
4. Acceptable deprecation period for member nodes = at least one year
5. Support multiple versions of types - need to name packages appropriately with version information
6. Review other package / apps for versioning strategies. e.g. Oracle, TLS, Google Maps, p2p networks? Amazon web service:

    http://docs.amazonwebservices.com/AmazonSimpleDB/latest/DeveloperGuide/index.html?APIVersioning.html

    http://stackoverflow.com/questions/389169/best-practices-for-api-versioning

    - good point. REST implies URL persistence, so we definitely need to support an unversioned URL, especially for gets on unversioned objects:

    http://blog.apigee.com/detail/api_restful_design_question_verisioning_number_in_the_url/

July 20 - Afternoon Session

Collating contact information for MN administrators and other stakeholders.

- Need a list of contacts to notify for:
    - version changes / system updates
    - system outages
    - other infrastructure related notifications

Contains:

- all member nodes: technical contact, administrative contact
- all "official" ITK components

DataONE needs one person who is responsible for notifications

primary contact as well as backup contact, user lead ("power user" perhaps)

Actions:

1. review MN registration doc (http://bit.ly/oDt9mz)
2. Setup mailing list with search capability
3. Identify someone responsible for sending notifications.

Public Release Design

Component Interface appearances

Use agreements and licenses

DataONE Terms & Conditions

- Review by CCIT - comments to Suzie within a couple of weeks - Bruce, Bob Sandusky

  In Google Docs https://docs.google.com/document/d/1TeVPzhsP53W-FwzFm3hl2J7EC5xjiNiSErd33Pyptno/edit?hl=en_US Send a note to Bruce Wilson (refbruce@gmail.com, bruce.wilson@utk.edu, or wilsonbe@ornl.gov) to get access. The intent is that Bruce and Bob will direct questions to CI and CE folks as appropriate.
- How to present use agreements?
  - No requirement to click through an agreement
  - Present links to use agreements where available (e.g. in mercury interface)
  - Goal is to provide information to users for "proper" use of the information - attribution, citation, redistribution, ... not as a legally binding agreement.
  - Member node agreements can contain a link to the Member Node data use agreement (presumably not conflicting with the DataONE operation)
  - Generally the best location for data use agreement / guidelines is within the science metadata for the dataset

Actions:

- generate a landing page that presents the use agreements
- Need a brief, general agreement for DataONE's perspective
  - Should say something like: "DataONE expects adherence to scientific principals on ethical data sharing, redistribution, and attribution. In addition, DataONE is a federation of data provider organizations each with their own usage policies and procedures, and we expect user's of data and information gathered from DataONE to respect the usage, redistribution, and attribution policies of the individual Member Nodes and contributors. It is incumbent upon data users to find the usage agreement information that are pertinent for all data downloaded. We provide the following links to the policies of individual Member Nodes to facilitate this process."
- Need links to the various MN specific links
- Need pointers to indicate that science metadata may contain more specific restrictions on use policies
- Who can do this?

What will be the face of DataONE at public release?

- e.g. using the GBIF site as a guiding structure. But that site is confusing to folks that want the data right away.

Front page:

- Need data search

- Other ways to access DataONE resources
- Links to how to participate
- Links to other aspects of the project
- Feedback - use a plugin feedback tool (commercial)

What are we calling the main page for the web interface to the DataONE CI?

Actions:

- Change the L&F of the docs.dataone.org site
- The header / menu bar (top section of page) should be designed such that it can be reused across multiple sites - drupal, Mercury, CILogon branded
- Design for the web site L&F elements needs to be at least in draft form for UI changes to Mercury
- Aim for consistency of presentation across all web interfaces for DataONE, perhaps with less detailed menus on search pages
- Giri will lead the web ui piece (search interface elements and design)
- Need a list of all Science Metadata formats
- For each format we need a transform to HTML

Implementation note: would it be prudent to add a visibility field to ObjectList (or ObjectInfo) to support marking individual results returned from search as "private" for example?

Searchable Elements

SearchMetadata.author:

```
(String)
Principle Investigator (PI) / Author
Name authority service is desired for some control over names appearing in metadata
For search, advantages to keep simple text representation
Goal of gradual increase in specificity
- Desireable if we can get "sort friendly" results -> recommend LAST, FIRST name
- Mercury UI
```

SearchMetadata.keyWord:

```
(String)
Keyword (uncontrolled keywords)

- Mercury enhances the kw list with e.g. gcmd keyword list
- Mercury UI
```

SearchMetadata.keyConcept:

```
Key concept -key concepts drawn from a set of ontologies
Term
Namespace
```

- Controlled lists of terms available for several metadata standards
- A topic being addressed by the semantics WG
- Unlikely to be available in 2011
- Giri has some keyword enhancement and mapping script (magic)

- Mercury UI

SearchMetadata.spatialFeature:

```
(SpatialFeature)
Spatial bounding box (largest bounding rectangle)
Spatial window (series of spatial envelopes representative of the spatial locations of where the data is collected from or relevant to)
Spatial features (points, bounding boxes, polygons)
Centroid
Bounding box
Polygon
(not largest extent)
(need to resolve the semantics of the bounding box search - e.g. if centroids are recorded but fall outside of bounding box search)
```

- Bounding box with contains or overlaps
- can be supported by SOLR and Metacat searches
- Mercury UI

SearchMetadata.namedLocation:

```
(String)
Named places
Term
Type
Context (Columbus OH, Columbus GA)
Namespace of gazetteer
```

- free text search field
- provide recommendations for representation (see Wieczorek's georeferencing document)
- Mercury UI

SearchMetadata.temporalCoverageStart:

```
(DateTime)
```

- temporal extent
- "jurassic" vs "date collected for jurassic specimen"
- not publication date, not creation date
- Applicability date (as opposed to collection date or publication date -- though searching on collection date is a secondary search). Multiple dates possible: Collection/observation date, coverage date, analysis date, publication date, metadata modification date (e.g. peat bog samples relevant to 50,000 - 10,000 BCE, collected in July 1980, re-analyzed in July 2008, published in January 2010, and metadata revised in June 2010). In this example, earliestDate applies to 50,000 BCE.
- Mercury UI

SearchMetadata.temporalCoverageEnd::

(DateTime) Temporal window Relative terms (e.g. terms from the Geologic time scale) need to be supported Date ranges

Temporal coverage of the data set (e.g. searches - during, before, after)

- same notes as above apply
- Mercury UI

SearchMetadata.any::

(String) Full text search / Text search on abstract - Mercury UI

Desirable

ElementsSearchMetadata.title::

(string) Title (2) - Mercury UI

SearchMetadata.objectFormat(String)::

Type of data (format) (2) Original data Summarized versions Method used for processing (to generate summary, or original data) Resource type (spatial, models, observations, web service, ...)

- search the system metadata value, but need to parse system metadata of related objects to determine the objectFormat(s) of the data.
- Potential for mapping the values appearing the science metadata to the controlled terms available in the object format registry
- label should be "content type"

SearchMetadata.variableName:

```
(String)
Scientific variables (from a controlled vocabulary) (1)
- not in Dryad, optional in most/all
- Mercury UI
```

SearchMetadata.dataDomain:

```
(String)
Domain of data (physics, environmental, ...) (1)
SCRATCH
```

SearchMetadata.scientificName:

```
(String)
Biological taxonomic extents (1)
- lots of opportunities for cleanup services from ITIS, EOL, etc
- Mercury UI
```

SearchMetadata.publication(String):

```
Search by publication (1)
"defining article"
- hard to define how to use this
- concept of the publication associated with generation of the data
SCRATCH
```

Some Others

SearchMetadata.submitter:

```
(String)
- for either data or metadata
- Principal that added the content to the MN
- Needs to be translated from the subject to the individual's name
```

SearchMetadata.relatedObject:

```
(String)
Related data (data sets, publications)
- given a PID find everything that refers to it (deprecated, describes, derivedFrom...)
- This should probably be specific to relation type - find all stuff derivedFrom PID
```

SearchMetadata.quality:

```
(String, controlled vocabulary)
Quality / level of curation
SCRATCH
```

SearchMetadata.relatedOrganizations:

```
(String)
Organizations involved in study
- low priority
```

SearchMetadata.size:

```
(Integer, long)
Size of data (bytes)
- needed for drive, display, not necessary for UI search
```

SearchMetadata.replicaCount:

```
(Integer)
SCRATCH
```

SearchMetadata.replicaLocation:

```
(String)
Number / location of replicas
SCRATCH
```

SearchMetadata.dimensions:

```
(Integer or perhaps float?)
Dimensionality of data
SCRATCH
```

SearchMetadata.measurementUnits:

```
(String)
Units of measure (for sci variables)
SCRATCH
```

SearchMetadata.identifier:

```
(Types.IdentifierType)
Identifier (PID)
- Need to be able to find metadata records describing a dataset identified by PID
- Find any science metadata that is or describes PID
- Mercury UI
```

SearchMetadata.datePublished:

```
- drawn from science metadata
- is this the same as publication date concept in datacite - Yes
- Mercury UI
```

SearchMetadata.dateAddedToRepository:

- first submission to Member Node (drawn from science metadata if available, otherwise system metadata)

SearchMetadata.dateSysMetadataModified(DateTime):

```
available from sysmeta
```

SearchMetadata.readPrincipal:

```
- element in permission index, used for shard query
```

SearchMetadata.writePrincipal:

```
(Types.PrincipalType)

Permissions on objects (e.g. available to read by user)
- element in permission index, used for shard query
```

July 21, 2011 Joint Meeting with Semantics Working Group and CCIT

Damian -- asked about Git. Brief discussion of the distributed storage working group.

Deborah -- rationalization of metadata? Key area of connection with the semantics working group.

Comment that the R demo is similar to examples with Maven.

Some thoughts on areas where need for input from Semantics WG:

(possibly secondary) Expand the "find similar data" function in Mercury.

Improving semantically enabled search

Bioportal ontology repository -- value and path forward. There has been work to implement a bioportal instance (d1sweb.dataone.utk.edu -- Line Pouchard lead). A concept is that this could become a specialized type of member node, working with semantic ontologies, providing identification, versioning, and curation services. These ontologies could also then be useful within the DataONE context for the overall semantic integration of data.

Categorization of keywords -- enabling science categories and science topics based on the structure and content of the science metadata. This is the "Key concept" notion that Matt mentioned, for expanding out the the DataONE Drive, as a better hierarchy for the drive. See comments below regarding the member node context that's relevant.

Expand out on the work from the keyword list provided from Dave. Represents the information from the datasets in the test list of datasets. Can we do something that creates what amounts to a piece of the CN infrastructure so that the Semantics WG has a read-only copy of the sharded metadata to provide some information about the context in which those keywords occur. This is SOLR indices. Note that there are different classes of keywords and data from different member nodes have different controlled vocabularies. There is a context piece of information that ties the member node source to some controlled vocabularies and relevant ifnormation. For example, the ORNL DAAC keywords will be drawn from the GCMD keyword list. KNB has a different controlled vocabulary. And both have the abiltiy to have both free-text keywords and controlled vocabulary keywords.

What kinds of tools are possible to allow users to type information in about what this data is about and have a sementic enablement that suggests appropriate keywords and topics based on the information supplied. This would be a semantic enablement in a different context of Morpho, for example. The reward for the user is better visibility of the data and potential for higher use (higher impact) of the data. Also of value for data centers that do a high level of curation (like the ORNL DAAC) where improving the metadata quality improves the overall impact of the data center.

What is the definition of data? Do we have this well defined in the architecture documentation? This would be a good element for what's in the reference architecture that still needs to be done.

The semantics and interoperability working group is keeping notes at:

> https://docs.google.com/document/d/1Vv6ekKh91oXtBWlRbxEjev2UurJuNnh1ygdqaPsTaRg/edit?hl=en_US

I (deborah) am also about to share this etherpad link with that page.

Sun use case (persona): Several questions. What is colocated data? What is colocated in the context of this problem (tortoise food web). Where is there some definition of tortoise food web. What met data is relevant (would have to take into account prevailing met patterns)?

Note that the fundamental difference between semantics that can be captured at point of data generation and semantics to be added to existing metdata. Begs for tools and best practices for semantics and markup at the point of data collection/generation.

Working Group Coordination: AHM and charters are a key issue. AHM meeting October 18-20, in Albuquerque. Will start Tuesday AM; finish late Thursday. About 45 minutes from ABQ. LT and CCIT should also have some role in cross-WG communication. WG's should feel free to communicate (telecon, webinar) for subgroups.

Mark Shildhauer -- Semantic Schmear or Semantic juju :-). for working on the keyword list.

The Questions from the I&S Working Group

- Is the keyword list still an issue that CCIT want the WG to address? Where is CCIT with this?
- Will D1 recommend a particular metadata schema for member nodes to use when they want to contribute their data to D1?
- Are Mercury and the metadata model a constraint or an early implementation of a low-hanging fruit?
- How much of this group is analysis and recommendation and how much should we actually be doing work?
- The degree to which D1 resources are accessible as Web resources from REST interface and/or w get resources?

The keyword problem

- ability to move from arbitrary list to an automated mechanism to annotate metadata (augment the search index) with key concepts may open access to a bunch of useful technologies
- context is key to leveraging power of what's availble in the semantics of science metadata
- context = "additional associated metadata"
- how to bootstrap semantic annotation of datasets beyond what is available in the science metadata
- often need to open the dat afile to discover additional information about the data set (variables, content range, ...)
- need to move from "Tair" -> "air temperature measured x m above ground using y instrument"
- There is reasonable structure in the metadata documents, though many of the element values are uncontrolled (e.g. Tair, air_temp, temperature, ...)
- metadata = EML, iso19115, dryad application profile, FGDC bio profile

- The process of extracting the metadata elements that are indexed for search is one place where we could enhance what we are doing now
- need to document use cases for discovery and use these to help drive the process of the Semantics WG
- There are existing use cases - need to capture these in one location

EVA group has two "science scenarios" being worked through:

A. the bird migration simulations. Lots of semantics issues wrt integration

B. climate change. Existing carbon models (input to IPCC) -> comparisons of predictions of the outputs to each other and to real observations to evaluate prediction of past events

    - sonet use cases, personas, ...

Need a trajectory for getting semantic technologies to work across the system - CCIT can help to get it into the infrastructure

Browse hierarchy and faceted search terms that can be used in search interface

```
extractKeyConcepts(metadata):

  - return a list of key concepts given a science metadata document
  1. identify context
  2. identify relevant resources (ontologies, ...) given context
  3. extract keywords from appropriate location in metadata document
  4. for each keword
     a) match keyword to key concept drawn from ontology / thesaurus / controlled source
  5. return list of concepts
```

An alternative approach is to say, given this list of key concepts, which are relevant to this science metadata document?

Need concise descriptions of what is required to implement key capabilities - then the CCIT team can allocate resources to get it done.

Possible Steps:

- Uncontrolled keywords (from metadata records) -> map to controlled keywords -> expand keywords using ontology lookup
- Uncontrolled keywords: generic keywords, theme keywords etc.. from the Science metadata records
- Controlled keywords: using CF variables list, GCMD hierarchical keywords, NBII Thesaurus etc..
- Ontology lookup using : SWEET, OBOE, DOLCE lite, ESG (model data..) Etc..
- Ontology portals: BioPortal, HIVE...

For ranking objects found in keyword searches, we could use a system like Google’s PageRank.

We know about object relationships:

- obsoletes, obsoletedBy -> object
- describes, describedBy -> object
- created by -> subject
- accessible by -> subject
- originating member node

Objects and subjects can be followed recursively to discover their objects and subjects. The discovered information can be used for adjusting the ranking of a given object for a given keyword. For instance, if an object has the keyword “water” in its science metadata, and is shared with subjects which have themselves created objects with the keyword “water”, the initial object would be given higher ranking for that keyword.

Idea: Putting additional metadata into System Metadata that allows subjects to point to objects that were of interest to them when they were doing research related to “this” object (to further help with keyword searches).

July 21 - Afternoon Session

Scheduling

Refer to spreadsheet:

https://repository.dataone.org/documents/Committees/CCIT/20110719_CCIT_SantaBarbara/docs/DataONE%20CI%20Components.xls

CN still needs Logging aggregation added.

OAI-ORE needs implementation on CNs, and /assertRelation service endpoint needs development

Debian packages need an update mechanism; configuration mechanism

Create a debian package for both Metacat and the GMN to install on a MemberNode instance

Later, we should distribute generic MemberNode VirtualMachine image for KVM and VMWare

For the spreadsheet (DataONE CI Components.xls):

> V1r1 = OAI-ORE add, sys meta changes, version support in package names, functionality updated to work with sys meta changes

Vacation time:

> Chris: Week of August 2 Robert: August 3-9 Rob: August 3-12 Roger: October Matt: July 25 week + 1 week Nick D: week of August 15