Overview NSDL Collection Representation and Information Flow

October 24, 2006

Introduction

The intent of this document is to be a brief overview of the current state of collections and associated processing/representation in the NDR. The hope is to convey a general understanding of the current collections architecture as it stands, as well as any advantages, limitations, and not-so-obvious design decisions that were made. It seems appropriate to start by explaining how data in collections currently gets out of the NDR and available to searches before delving further into modeling details.

Data Out: OAI

The existing NSDL infrastructure centers around our old metadata repository (MR), and for all practical purposes the only way to gather information from the MR is OAI. Since the OAI data available from the MR is inherently metadata-centric by design, the first incarnation of OAI available from the NDR will also be metadata-centric. Thus, currently the only way to take advantage of the NDR's more flexible data model is to use the API directly. As most of the NSDL's core services still assume the metadata-centric OAI paradigm, the metadata-centric OAI served by the NDR is likely to remain the largest public exposure of NDR content (by volume) for the near future.

Metadata-centric OAI from the NDR

The current set of rules for determining the OAI content served from the NDR is as follows:

1. Any non-deleted object with an `itemID` property will be considered eligible for serving out via OAI. Of the items with an `itemID` property, only those with a `getMetadata` dissemination will be considered.
    - All metadata objects inherently have a `getMetadata` dissemination defined for them.

- The `getMetadata` dissemination of each metadata object looks for `format_nsdl_dc` and `format_nsdl_dc_info` datastreams. This dissemination applies an XSLT transform, and perhaps some additional processing, in order to produce output in a different metadata format (currently nsdl_dc, oai_dc, nsdl_search), which is specified as an argument to the dissemination.
- The NDR's OAI service periodically polls the NDR for new/deleted records that contain the requested disseminations in the requested format.
- The `getMetadata` dissemination content cached and managed by Fedora's oaiprovider service, and is served to the public via OAI.

2. If the matching object has a `metadataProvidedBy` relationship with a MetadataProvider object that contains `setSpec` and `setName` properties[1], then that piece of metadata will appear in the OAI set designated by `setSpec`.

..and that's essentially the entire process. There are a few important implications that should be clear:

- Most of the NDR data graph is invisible to OAI: only Metadata objects and, optionally, MetadataProvider objects are seen by OAI.
- Of the Metadata objects, only those that have an nsdl_dc datastream can currently disseminate content
- The `getMetadata` disseminations would have to be modified in order to generate content from something other than nsdl_dc or to something other than nsdl_dc, nsdl_search or oai_dc
- Disseminations are not currently exposed in the API

Searching NDR contents

The NSDL search service allows discovery of resources by matching text found in metadata fields or the fetched content of resources. The existing search service in production is a metadata-centric search in that it returns all matching metadata records. "Search 2.0" is a resource-centric search, in that while it still searches the content of metadata records, the results are given in terms of distinct resources that have matching metadata or textual content. That is to say, each "hit" in a the search result corresponds to a resource rather than a specific metadata record. The input to Search 2.0 is currently a metadata-centric OAI feed, so its resource-centric processing is completely independent of the capabilities of the data source. Thus, in order for NDR content to be visible to the NSDL search service, it must be visible via OAI.

[1] currently, exposed in the API as existing in the default request namespace. We are considering changing this.

As far as indexing content of resources, Search 2.0 gathers resource URLs from the metadata contents and crawls content online

The fact that the NSDL search index is built from OAI carries some clear implications on what is currently searchable in the NDR from that service: metadata that has an nsdl_dc datastream. All other content in the NDR is invisible to the search service

Collection Definition & Modeling

The figure at the end of this document depicts the representation of a collection in the NDR. In this figure, we represent collection content (composed of metadata and resource items present in the collection, the collection definition (the set of 'administrative' objects that define the collection's representation in the NDR), and the management of collections (identifying and describing a set of objects in the NDR as a collection). The definition of a collection is perhaps the least obvious aspect of our collections model, and thus will be described in the most detail.

A collection is represented by a set of at least five NDR objects and a "specialized" relationship.

Collection Agent represents the source or owner of a particular collection (e.g. DLESE, NEEDS). Currently, most collection agents are placeholders determined by an automated process, and may or may not represent the true owner/source of a collection. The intent is that as providers gain the ability to control their content in the NDR, the collection agent would serve as their identity when authenticating and performing authorized actions, such as adding/deleting items from one of the collections they own.

Collection Aggregator Defines the set of resources in the NDR that are in a given collection. For collections managed by the NSDL's harvest/ingest process, the aggregators will contain a `crs:collectionID` property containing an internal collection identifier[2]

Collection Metadata Provider Defines a set of metadata in the NDR from the Collection's Agent that describe resources that are in the collection. Some important properties of collection MetadataProviders

- The MetadataProvider contains `setSpec` and `setName` properties that designate the metadata set in OAI
- MetadataProviders whose metadata is imported into the NDR via OAI Harvest/Ingest process have an administrative property called

[2]This property, which exists in the `http://ns.nsdl.org/ndr/collections#collectionID` namespace, is a 'private' property in the sense that it is not part of the core NSDL model, nor part of the general definition of a collection. It is an implementation detail of our collection management application, which is why it is placed in a different namespace

`crs:collectionNA`, which serves to identify the collection to our ingest process. They also contain a `harvestRequest` datastream, which contains information related to harvest processing/scheduling.

Collection Resource For a collection that exists as an online resource outside of the NDR (as just about all currently do), the collection Aggregator is associated with an NDR resource that represents that online presence. For example, the KMODDL collection contains a number of items. In its collection definition, the collection resource corresponds to `http://kmoddl.library.cornell.edu/`, which is that collection's homepage.

Collection Metadata Describes the collection. Collections in the NDR are distinguished as collections by the membership of their Aggregator in the 'NSDL Collections' aggregation. Thus, the 'official' metadata that describes the collection describes the collection *Aggregator*. Technically, the collection resource and the collection Aggregator are distinct and should deserve their own metadata. In practice, however, we assign the same metadata to both the collection resource and the collection Aggregator. Since we give them identical metadata, we end up creating only a single metadata object with `metadataFor` relationships for both the collection Resource and the collection Aggregator.

`aggregatedBy` **relationship** Relationship specific to collections. It signifies a contract between a collection's Aggregator and MetadataProvider that states that all the metadata provided by the MetadataProvider describes resources in the Aggregator.

Some key points about collection definition in the NDR are:

- A collection agent may own many different collections. As an example, if DLESE has three collections in the NDR, its agent would have three Aggregators and three MetadataProviders attached to it.
- Some collections in the NSDL do not have items. These are typically online collections that do not provide OAI. This does not affect how they are defined in the NDR (i.e. they still have collection Aggregator, MetadataProvider, etc, even though these may remain empty).
- Existing NSDL collections have some administrative properties and datastreams (such as harvestRequest) that are really only useful to our ingest process, and probably will not be defined for collections populated by other means or organizations.

The representation of collection content and collection management, in contrast to collection definition, are relatively straightforward as can be seen from the figure. There are, however, a few things to note:

- There does not have to be a 1:1 relationship between collection metadata and resources. Some collections have multiple Metadata for a single resource, or a single Metadata object that describes multiple Resources.
- The 'contract' implied by the `aggregatedBy` relationship is not currently enforced by the NDR API. Its intent was mainly to be a marker, but we may consider enforcing some logic in the future, if there is desire to do so.
- Permission to add collections to the NSDL Collections Aggregator or add metadata provided by the NSDL Collections MetadataProvider is restricted. Any user of the NDR who intends on adding official collections to the NSDL must get permission (i.e. have their Agent authorized) to do so.

Data In: Ingest

Currently, every item in the NDR (and MR) that is in a collection originates from an OAI harvest from some external source. As individual content providers such as DLESE begin using the NDR API to insert collection data, this statistic will change. At the present, however, it remains true that the bulk majority of collection data in the NDR is from our in-house OAI harvest/ingest process.

Because the source material for collection content is currently OAI-harvested metadata, our ingest process needs to derive resources and some administrative collection information solely from this given metadata. As such, the procedures used in the NSDL ingest/harvest process will probably be quite different from those that are intended for content providers who push data directly into the NDR. Content providers will likely have much more information about their resources and metadata on hand, and will not necessarily have to derive all their data indirectly from OAI harvests. Despite these differences, an overview of our ingest process is useful for illustrative purposes.

Our harvest/ingest process is as follows. Currently, it employs some functionality that is not exposed by the NDR API, such as direct graph searches. Any functionality that is not currently afforded by the NDR API is in *emphasized* print:

1. Metadata is OAI harvested from a source and parsed to an intermediate, internal format that is used by the ingest process to form the actual API requests
    - Besides actual metadata content, this internal format contains a 'crs:collectionNA' value, which represents that collection's identity in our collection management system (CRS)
2. Scan the metadata for `<dc:identifier>` element. From each `dc:identifier`, create a Resource object. Determine if a resource already exists with that identifier[3]:

[3] using the findResource API method. If the resource is a URL, it will be normalized.

(a) If a resource already exists, modify that resource and add it to the collection's Aggregator

(b) If a resource does not exist, create a new one, and add it to the collection's Aggregator

3. When processing metadata in this internal format, each metadata record is mapped to a NDR MetadataProvider object *by querying for the MetadataProvider with a matching* `crs:collectionNA` *property.*

4. One or more 'format' datastreams are prepared for each metadata record from the source metadata content.

   - The exact metadata from the source record is prepared as a 'native' metadata format datastream, for example oai_dc from a source record would be put into a datastream such as `format_native_oai_dc_v1.0` This native datastream exists so that the NSDL retains a copy of the source metadata in its original format.
   - An nsdl_dc representation of the metadata is prepared, possibly as a transformation of the incoming metadata format. As discussed earlier, currently, metadata must exist in nsdl_dc in order to be visible by the the NDR OAI and search service. Thus, this content is formed into a datastream named `format_nsdl_dc`

5. An 'info' datastream is prepared for each metadata record. The contents of this datastream will form the basis for the 'about' section in any metadata served out by the NDR's OAI service.

   - A required element of the info datastream is `<link>`. The contents of this is NSDL OAI identifier of the 'parent' of the resource described by this metadata. In other words, the OAI identifier of the metadata describing the collection containing the resource described by this metadata record. In any case, the ingest process must find the `itemID` of the Metadata that is `metadataFor` the Collection Aggregator, which is related to the Collection MetadataProvider by the `aggregatedBy` relationship. This is done by *querying the graph*[4].

6. For each metadata record to be inserted, check to see if it already exists in the NDR for the purposes of determining if an `add` or `modify` operation is needed

   - A metadata record can be determined to exist by *searching for matching metadata records with the same* `uniqueID` *property and* `metadataProvidedBy` *relationship.*

[4]This `<link>` element in the info datastream currently performs an important function in the context of the NSDL search service. Since the search service only knows what is exposed through OAI, the `<link>` element in the metadata's 'about' section is the means through which a metadata record declares that it is a member of a particular NSDL collection. This requirement may change once the search service polls the NDR directly for resource-centric data.

- The ingest process uses the originating source OAI identifier as the `uniqueID` property value[5]

7. Determine an `itemID` to be used as the OAI identifier of the metadata record

8. Finally, with all content prerequisites met (i.e. known `metadataFor`, `metadataProvidedBy` handles, `uniqueID` and `itemID` properties, and correctly formed 'info' and 'format' datastreams, add or insert the metadata item as appropriate

As is evident, our in-house ingest process is quite complicated, and suffers a few inherent drawbacks, such as having to derive the resources solely based on `dc:identifier` content. Thus, it should be clear that having content providers manage their data directly in the NDR has certain advantages.

Some changes that are on the horizon include (1) the search service may poll the NDR directly for searchable content and (2) The NDR OAI service may able to serve out resource-centric metadata, which is derived from from the contextual knowledge surrounding a resource in in the NDR graph. These changes are not finalized yet, but would perhaps relax some of the current requirements and conventions related to data flow into and out of collections.

Collection Management

All collections in NSDL are currently managed through the Collection Registration Service (CRS). This system performs the following critical collections-related functions, among others:

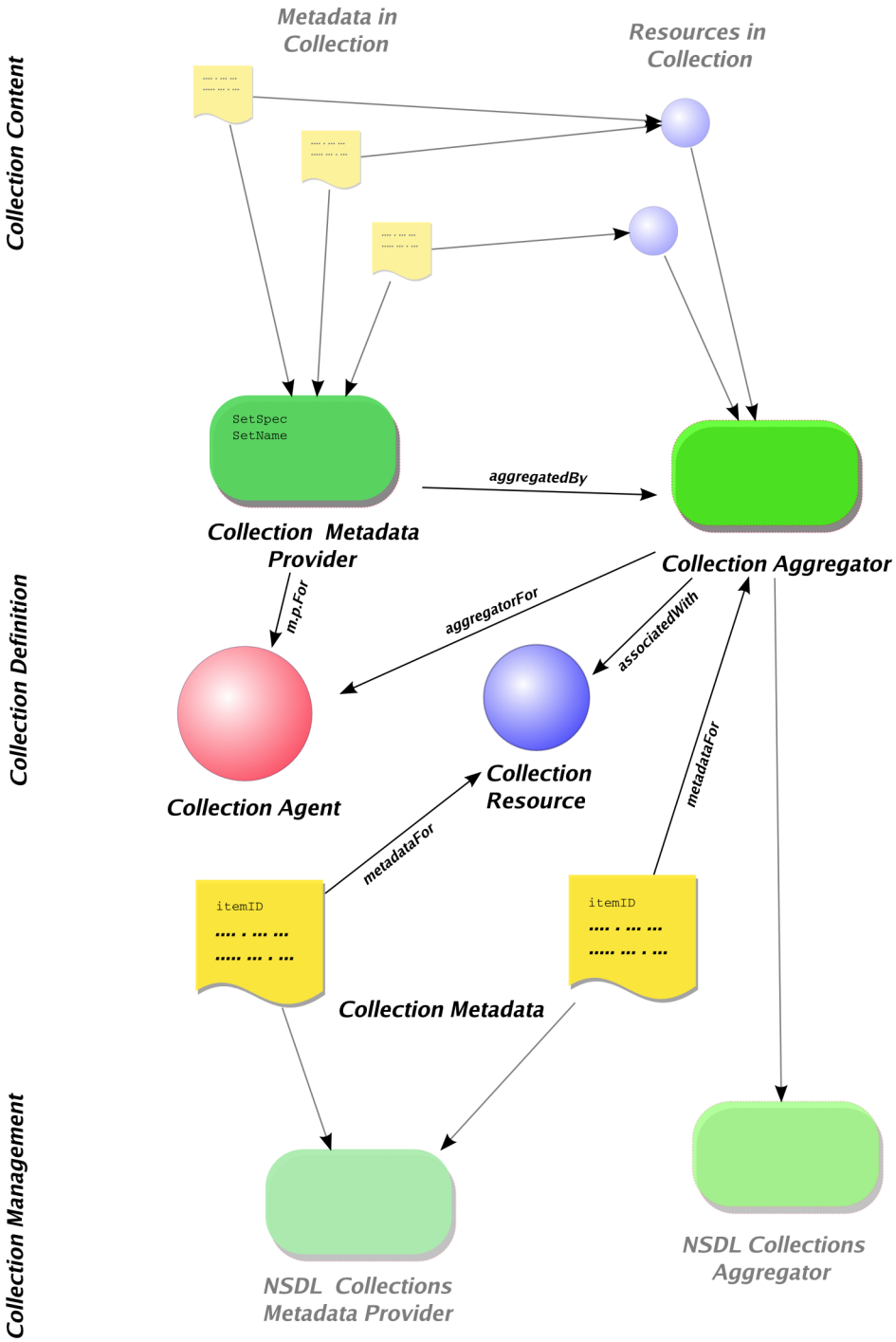
- Define metadata that describes a collection
- Add, delete, and modify collections in the Metadata Repository (MR)
- Assign identifiers to and keep track of all collections and collection data, including collections that have been deleted and those that are 'in progress' (i.e. not in the MR yet)
- Define and schedule OAI harvests for collections that support OAI

The CRS is very tightly integrated with the MR and its Oracle implementation, and is completely independent of the NDR. At the present, collection data is imported into the NDR from the MR in an external synchronization process that scans the CRS, compares collection information from the NDR, and makes appropriate updates to the NDR. Thus, the collection Metadata, Agent , Resource, and `crs:collectionNA` property in MetadataProviders are derived from the CRS.

[5]This implies that the ingest process must be careful of a particular vulnerability: a source changing its OAI identifiers. Complete reharvests (i.e. deleting all records in a collection and adding all harvested records anew) do not have this vulnerability

Eventually, collection management in the NDR will have to be handled by a more 'native' and sustainable solution, and some questions may have to be addressed, such as

- How will external content providers who directly control their data in the NDR interact with a centralized collection management service, if at all?
- What requirements on collection representation would a new collection management system impose?
- How will content providers transition from having their collection content harvested and managed by the NSDL to providing and managing their collections themselves under the new collection management system?


Collection Content
Collection Definition
Collection Management
Metadata in Collection
Resources in Collection
SetSpec
SetName
aggregatedBy
Collection Metadata Provider
Collection Aggregator
m.p.For
aggregatorFor
associatedWith
Collection Agent
Collection Resource
metadataFor
metadataFor
itemID
itemID
Collection Metadata
NSDL Collections Metadata Provider
NSDL Collections Aggregator