Report to ESIP Federation Data Stewardship Committee on the Implementation of Selected Identifier Schemes to Earth Science Data Objects as part of its Identifier Testbed Activities

Draft of 6 January 2014

By Nancy J. Hoebelheinrich, Knowledge Motifs LLC & Greg Janée, University of California at Santa Barbara

Executive Summary: In hands-on testing the ESIP Federation's Data Stewardship Committee Identifier Testbed Project found that Digital Object Identifiers (DOI), Archival Resource Keys (ARK), and Handles are the most suitable identifier schemes to serve as unique locators for Earth system science data and information. Implementation of these three schemes demonstrated their:

- **ease of use**, both for human users in creating and managing individual identifiers and for programmatic clients in operating on batches of identifiers via APIs;
- **support**, in terms of documentation and infrastructure, and of the stability of the backing organizations;
- widespread **adoption** by different communities;
- **scalability** with respect to both the number of identifiers that may be created and the rapidity with which they may be created.

DOI and ARK identifiers additionally offer support for citations, and publishers have expressed interest in harvesting these identifiers and their citation metadata into search systems. Handles require more investment, particularly in that the client organization must run a specialized, dedicated, local server. By contrast, DOI and ARK identifiers are accessed via centralized, global systems.

The ID Testbed Project found that Universally Unique Identifiers (UUID) are most appropriately used as unique identifiers, and are the most appropriate scheme for that purpose. Life Science Identifiers (LSID) are possibly suitable as unique identifiers, but they are unsuitable as unique locators due to their incorporation of domain names. Additionally, LSID suffers from relatively low adoption by a narrow community.

Persistent Uniform Resource Locators (PURL) offer no means for creating opaque identifiers, and the API support for batch operations is poor. Object Identifiers (OID) and Extensible Resource Identifiers (XRI) are least suitable for Earth system science data. The XRI scheme has yet to become fully operational. OID identifiers are really targeted at other information types (controlled vocabulary terms, for example).

Introduction: Since its inception as the ESIP Federation's Preservation and Stewardship Cluster and in its current incarnation as the Data Preservation Committee, one of the group's primary objectives has been to support the long-term preservation of Earth system science data and information. Both the Cluster and the Committee have provided a forum for ESIP members to collaborate on data preservation

issues. One of the areas in which the Committee has been quite active is in the exploration of persistent identifier schemes for Earth science data products, knowing that unique and lasting identifiers for data are needed for a wide range of purposes at various levels of granularity, from individual files, to data sets, to collections of data sets. Along with the NASA Technology Infusion Working Group, the ESIP Data Stewardship Committee has recognized that data centers storing, managing and distributing data products will need to support a variety of identification schemes and identifiers. To satisfy the requirements for identification purposes, a working group identified a number of identification schemes, each of which could potentially satisfy a subset of the overall needs. The identification schemes investigated include Digital Object Identifiers (DOI), Archival Resource Keys (ARK), Persistent Uniform Resource Locators (PURL), Handles, Object Identifiers (OID), Universally Unique Identifiers (UUID), Extensible Resource Identifiers (XRI), and Life Science Identifiers (LSID).

The purpose of the ESIP Data Stewardship Identifier Testbed (ID Testbed) activity has been to test and demonstrate the applicability of the selected identifier schemes to a variety of Earth Science data types with the ultimate goal of recommending a suite for use by ESIP Federation members. Following the work of the Data Stewardship Committee to identify and evaluate the eight identifier schemes with the most potential for assignment to Earth Science data, our task has been to explore and report upon the operational issues associated with the implementation of each of the identifier schemes to two different types of canonical Earth Science data sets, and their components.

Problem Statement: Given the features of each of the eight identifier schemes that caused them to be considered as candidates for recommended use by the ESIP Data Stewardship Committee, and the results of the initial assessment, what additional operational factors come into play upon implementing a given scheme that might change or affirm the Committee's recommendations?

Testbed Datasets: To address this question, the ID Testbed team investigated and implemented, when feasible, each of the identifier schemes in a test environment, assigned identifiers to two data sets, and when possible, to one or more components of each dataset. The data sets chosen were:

1) The Glacier Photo Collection from the National Snow and Ice Data Center, a photographic image collection in JPEG and TIFF format, [1] and

2) A numerical data set that is a subset of Level-2 and Level-3 data products from the Moderate Resolution Imaging Spectroradiometer (MODIS) sensor on NASA's Earth Observing System (EOS) Aqua and Terra satellites representing snow cover and sea ice. [2]

The purpose of implementing each identifier scheme and assigning identifiers to canonical data sets and objects or granules within data sets was to more fully understand the kinds of decisions that would need to be made in order to address the requirements of four different use cases associated with a data set

[1] http://nsidc.org/data/glacier_photo/

[2] See the Overview for these data products at: http://nsidc.org/data/modis/.

and its components. The four use cases explored unique identification, unique location, citable identification, and scientifically unique identification for a data set and its components.

Background Information / Context:

A paper written by members of the ESIP Data Stewardship Committee, *On the Utility of Identification Schemes for Digital Earth Science Data: An Assessment and Recommendation,* (Duerr *et al*, 2011) set the stage for the ID Testbed by discussing the purposes for identifier schemes and characteristics of identifier schemes that seemed most appropriate for Earth Science data sets. The four key purposes for which identifier schemes are used for data are to:

1. Unambiguously and uniquely identify or differentiate one data object from another
2. Access and find data without regard to its current location or system of management
3. Facilitate the management of data objects over time
4. Facilitate data citation in traditional publications

In order to assess the extent to which the different identifier schemes could meet the purposes highlighted, the paper described each scheme in terms of its naming syntax, technical value, user value, value to archives or data centers, and capability for supporting the following four use cases:

- Unique Identification: To uniquely and unambiguously identify a particular piece of data, no matter which copy a user has
- Unique Location: To locate an authoritative copy of the data no matter where they are currently held
- Citable Location: To identify cited data
- Scientifically Unique Identification: To be able to tell that two data instances contain the same information even if the formats are different

The Duerr *et al* paper discusses each use case more fully, but for our purposes we identified key requirements that relate to the operational issues for each use case. In the process of implementing the eight identifier schemes, we attempted to evaluate the capabilities inherent in each of them to assess whether they could meet the requirements of the use cases from an operational point of view. The following chart describes the requirements and the use case(s) to which they apply:

Requirements by Use Case per Duerr et al Identifier Paper

	A	B	C	D	E	F
1	**Paper Requirement**	**Means for testing in Testbed environment**	**Unique ID use case?**	**Unique locator use case?**	**Citable ID use case?**	**Scientifically unique Use Case?**
2	Location "independence" **	Is a property of an identifier. Preferably, would be embedded in the data object or be computed, like a checksum. Did not test.	x			x
3	Location "invariance" **	Is a property of an identifier. Would be able to ascertain if the DO could be found regardless of its current location. Did not test.		x	x	
4	Generated at time of DO creation	Restrictions on when DO can be created? Who can create? Can be created in the field? For testbed, DO already created. N.A. except for process of assignment of IDs.	x			x
5	Can be created after DO is considered to be permanently available	Who can create? Data producer or data archive? Can be created in the field? For testbed, DO already created. N.A. except for process of assignment of IDs.		x	x	x
6	ID necessarily placed within or carried along with DO	Not practical to test; is a practice based on usage.	x			x
7	Referenced with descriptive MD for DO	Is descriptive MD associated with the DO? If so, how extensive and how associated.	x			x
8	Difficult to change once established	Test to see if we can go back & change the ID once it's created	x			x
9	Created once & never modified thereafter	Test to see if we can go back & change the ID once it's created	x			
10	Globally unique	Property of the identifier. Assess what mechanisms within the ID provider system or service organizations exist to verifiy global uniqueness? Is there a way to check for duplicates within the ID provider system?	x	x	x	x
11	No requirement for 3rd party naming authority beyond data producer	Ck for operational rules associated with account / ID creation, if applicable.	x			x
12	3rd party naming authority considered necessary & useful	Ck for operational rules associated with account / ID creation, if applicable.		x	x	

Requirements by Use Case per Duerr et al Identifier Paper

	A	B	C	D	E	F
13	Data is findable after creation despite changes in custodianship	Availability of landing page? Not tested in the wild since the sources of the data sets we used are trusted. Is a recommended practice.		x	x	
14	Points to trusted source that can authenticate current copy	Availability of landing page? Not tested in the wild since the sources of the data sets we used are trusted. Is a recommended practice.		x	x	x
15	Mechanism exists for locating earlier & superceded versions of DO	Availability of landing page? Not tested in the wild since the sources of the data sets we used are trusted. Is a recommended practice.		x	x	x
16	Mechanism exists to perform ongoing maintenance of ID	Tested by changing descriptive MD after initial establishment. Where is that done & who has the responsibility for initiating? How are changes conveyed (e.g., batch or one by one?)		x	x	
17	Broadly accepted by publishers	Check on web page to see if used; or contact science publishers, e.g., HighWire Press, Thompson-Reuters, others?			x	
18	Can be assigned to data sets	Test ease of doing & how easily referenced by machines & humans		x	x	
19	Can be assigned to components of data sets	Test ease of doing & how easily referenced by machines & humans		x	x	
20	Has mechanism for intrinsic verification of semantic equivalence of data content	Only way to assess by noting how scheme differentiates b/w different formats of same content? A mechanism such as this would need to be not only an identifier, but also something like a checksum or message digest that would perform other functions related to differentiation or semantic equivalence (e.g., with statistical data that can be re-formed, but still has the same semantic meaning).				x

Methodology

Generally, the methodology for the Testbed activities was the following:

1. Identify data sets / components for the Testbed

2. For each ID scheme, research / explore options for organizational assistance in assigning IDs to each data set and its components such as service providers, standards organizations, etc. Choose ID provider.

3. Establish accounts with chosen ID provider.

4. Research / explore associated rules, and options for assigning IDs.

5. Go through process of assigning IDs; record observations and answers to operational questions, listed below.

Questions asked during testing process:

As previously described, the objectives of the Duerr *et al* paper were to provide an initial assessment of the identifier schemes that might be most feasible to use for Earth Science data sets rather than to assess how the schemes might be implemented and used in real world situations. Recognizing that the operational issues arising out of implementation might be important factors to help an organization decide which identifier scheme would be most appropriate for its data, the ESIP Data Stewardship Committee included the recommendation that the Committee's initial list of chosen ID schemes should be tested. In order to test the implementation of the eight schemes, implementers of the ID testbed addressed questions that were slightly different, but related to the questions initially asked. (See chart above.

The operational questions asked of each identifier scheme were the following:

Category	Questions Asked
ID Assignment / Maintenance Issues	1. What is relationship with URI? (Addresses unique identifier capability, and interoperability)
	2. What is relationship with URL? (one of the main citable locator schemes)
	3. What is relationship with URN? (as internet naming scheme)
	4. What are the rules, roles & rights for roles to assign IDs, delete IDs, edit IDs or associated MD?
	5. Is it possible to batch create identifiers? If so, how done? (Addresses scalability)
	6. Is a registry or registration service required or provided? (Addresses third party maintenance)

	7. If a registry service is available, what are the services provided?
	8. Is a specific naming scheme and/or naming authority required?
	9. What is the ID for the data set level digital object? For the component level digital object?
	10. What is the technical infrastructure upon which the implementation of the ID scheme is based, (e.g., XML, Java, Python, other)?
	11. What is the mechanism for ensuring uniqueness of the ID within the ID provider system and/or globally?
	12. Is there a way to declare or describe a relationship between / among different formats of the same intellectual content within the ID itself? (Addresses semantic equivalence)
Discovery Issues	1. Who is using the scheme and for what purpose(s)?
	2. If used as citation for publication, are the citations counted by aggregators (e.g., ISI or Thomson's Web of Science)
	3. Is it possible or recommended that descriptive metadata be associated with the ID? If so, how maintained?
	4. Can the identifier have semantics? If so, what part / how?
Archival Issues	1. How is the association between the ID and the resource maintained when transferred from one archive or repository to another (e.g., embedded within object?)
	2. What are the outright charges or costs involved in assigning the IDs or using an associated service? (initial and ongoing on annual basis)
	3. What kind of staff resources are required to implement the scheme, associated software or service? (type, knowledge needed)
	4. Does the identification scheme have a mechanism for association with related data objects, such as physical documentation or an associated spreadsheet? Does it handle data that is not on the web? What about physical objects?

A spreadsheet detailing the specific answers to these questions for each of the identifier schemes is attached to this report as **Appendix A: Identifier Scheme Comparison Table**.

Findings:

Due to the interest in using identifiers in multiple web-accessible applications such as citation and data object location services, we found it necessary to gain some understanding of the relationship of each of the ID schemes to the primary web based technologies (URIs, URLs & URNs) for identifiers.

URI: Uniform Resource Identifier (RFC 3986) provides an extensible means for identifying a resource within the World Wide Web. Each URI begins with a scheme name that refers to a specification for assigning identifiers within that scheme; each scheme's specification may further restrict the syntax and semantics of identifiers using that scheme. URI specification defines (1) an implementation to access a location on a file server, commonly accessed using the http protocol though other protocols are allowed; (2) a syntax for referencing through which other ID schemes can be specified as URIs, e.g., ISBNs. The network path of the URI is implicitly DNS based; original URI specifications that assume the URI to be opaque have been overtaken by practical usage which assumes that the initial URI parser will look for meaningful characters (such as dot and slash).[3]

URN: Uniform Resource Name (RFC 2141) is a specification for defining names (identifiers) of resources for use on the Internet. Locations are assumed to be independent of names. URN architecture assumes a DNS-based Resolution Discovery Service (RDS) to find the service appropriate to the given URN scheme. However no such widely deployed RDS schemes currently exist: browsers cannot action URN strings without some additional programming in the form of a "plug-in". The set of URNs, of the form "urn:nid:nnnnnn", is a URN namespace. [4]

URL: Uniform Resource Locator (RFC 1738) is a location on a file server in the WWW; redefined in RFC 3986 as "a type of URI that identifies a resource via a representation of its primary access mechanism (e.g., its network "location"), rather than by some other attributes it may have". In this view "URL is a useful but informal concept" (RFC 3305). In practice, it identifies a single location, and therefore is usually used incorrectly as an identifier of the resource at a particular location. In addition, URLs carry semantics of the Domain Name they are based on and are therefore unsuitable as opaque identifiers; they may also be contextually qualified. URLs are pervasive as the foremost mechanism of location specification throughout the WWW, but less useful as persistent or unique identifiers. [5]

Digital Object Identifier (DOI)

Relationship of DOI to URI/URL/URN

URI: DOIs can identify physical objects as well as digital objects and may be used in applications other than the WWW or Internet. The syntax for Uniform Resource Identifiers (URIs) is much more restrictive than the syntax for the DOI. DOI is registered with info URI scheme (RFC 4452: http://info-uri.info) that

[3] http://www.doi.org/factsheets/DOIIdentifierSpecs.html

[4] Ibid.

[5] Ibid.

was developed by library and publishing communities for "URIs of information assets that have identifiers in public namespaces but have no representation within the URI allocation".[6]

URN: DOI is not registered as a URN namespace, despite fulfilling all the functional requirements, since URN registration appears to offer no advantage to the DOI System. It requires an additional layer of administration for defining DOI as a URN namespace (the string urn:doi:10.1000/1 rather than the simpler doi:10.1000/1) and an additional step of unnecessary redirection to access the resolution service, already achieved through either http proxy or native resolution. If RDS mechanisms supporting URN specifications become widely available, DOI will be registered as a URN. [7]

URL: Hexadecimal (%) encoding must be used for characters in a DOI that are not allowed, or have other meanings, in URLs or URNs. *Note:* The DOI itself does not change with encoding, merely its representation in a URL. A DOI that has been encoded is decoded before being sent to the DOI Registry. At the moment the decoding is handled by the proxy server http://dx.doi.org/. Only unencoded DOIs are stored in the DOI Registry database. For example, the number above is in the DOI Registry as "10.1000/456#789" *and not* "10.1000/456%23789". The percent character (%) must always be hex encoded (%25) in any URLs.

A DOI name may be represented as a URL (http string) by prefacing the string http://dx.doi.org/ to the DOI of the document (e.g., to resolve the DOI name 10.1000/182, enter into a browser the address: http://dx.doi.org/10.1000/182). Web pages or other hypertext documents can include hypertext links in this form. [8]

Implementation Specifics:

Digital Object Identifiers (DOIs) are administered by the International DOI Foundation, which hands off the work of actually registering identifiers to "registration agencies." There are only a few such agencies, CrossRef being the most well-known due to its role as the primary DOI registrar for journals and print publications. For this testbed work we chose DataCite, a relatively new registration agency focused on identifying datasets. DataCite in turn hands off registration to its member organizations, or "allocators." We used the California Digital Library allocator and, specifically, its EZID service at http://ezid.cdlib.org. [9]

Getting started with the EZID Service requires establishing a contractual relationship with the University of California and paying an annual subscription fee that is dependent on the nature and size of the applying institution. (DOIs were once charged per identifier, but no longer.) Once set up, EZID provides both a web user interface for creating identifiers by hand, and an API for batch operations.

---

[6] RFC 4452: http://info-uri.info. For more information, see "DOI® System and Internet Identifier Specifications".

[7] For more information, see "DOI® System and Internet Identifier Specifications".

[8] Ibid.

[9] Disclosure: one of the authors (Janée) is the principal developer of the EZID system.

We created collection-level identifiers for a MODIS collection, doi:10.5060/D4CC0XMZ, and the NSIDC Glacier Photo Collection, doi:10.5060/D4RN35SD. Notice the opacity of these identifiers: they were generated automatically by EZID (custom identifiers can also be created). These identifiers resolve through the DOI resolver at dx.doi.org by appending the identifier, as in http://dx.doi.org/10.5060/D4CC0XMZ. Identifier metadata can be viewed within EZID by appending the identifier to EZID, as in http://n2t.net/ezid/id/doi:10.5060/D4CC0XMZ.

We found that the ability to assign granule-level identifiers is dependent upon the nature of the data set. For example, the MODIS data set did not lend itself to assigning granule-level identifiers because, for that data set, there is no one web resource representing a granule. Instead, individual aspects of a granule (data, metadata, etc.) are available through different URLs, all of which are tied together by the NSIDC data access portal.

For the Glacier Photo data set, a more loosely connected collection of thematically related, yet independent, items, we considered three naming schemes for assigning identifiers to individual members of the collection:

- An opaque identifier for the granule. This requires that the collection maintainer somehow create and manage a database of identifier-granule mappings.
- An identifier that consists of the granule's local identifier appended to (or otherwise embedded in) a root or base identifier.
- A single base identifier that supports partial resolution. DOIs do not support this approach but, as will be seen later, PURLs do.

We implemented the second approach, creating, for example, granule identifier doi:10.5060/D4RN35SD/baird1929090101 for the photo with local ID baird1929090101.

DOIs created through DataCite (the registration agency EZID uses) require metadata, and a DataCite has developed a metadata schema that is intended to support journal-like citations and more. We mapped collection-level metadata to DataCite's schema with little difficulty. By request of the collection maintenance staff at NSIDC, we did *not* assign more than the minimally required metadata (title, creator, publisher, publication date) to the Glacier Photo Collection since the assignment of additional metadata was seen as an unnecessary management burden to maintain.

Archival Resource Key (ARK)

Relationship of ARK to URI/URL/URN

URI: An ARK is a URL, and so can be considered a URI. An ARK can be assigned to any type of object:

- digital objects such as documents, databases, images, software, websites, etc.;
- physical objects such as books, bones, statues, etc.
- living beings and groups such as people, animals, companies, orchestras, etc.

- intangible objects such as places, chemicals, diseases, vocabulary terms, performances, etc.

URL: An ARK is a URL, but has specific syntax requirements for the "label" part of the URL. [http://NMAH/]ark:/NAAN/Name[Qualifier]. The NAAN is the Name Assigning Authority Number - mandatory unique identifier of the organization that originally named the object; the NMAH is the Name Mapping Authority Host - optional and replaceable hostname of an organization that currently provides service for the object; the Qualifier is an optional string that extends the base ARK to support access to individual hierarchical subcomponents of an object, and to variants (versions, languages, formats) of components.

URN: URNs are designed to describe an identity rather than a location; thus, because an ARK as a URL specifies a location (along with an implied commitment to persistence by virtue of its specific syntax), it is not a URN. Also, URN namespace assignments are handled via the IANA, the Internet Assigned Numbers Authority. Namespaces for the ARK are maintained as part of the NAAN Registry that is maintained by the California Digital Library and replicated at the Bibliothèque Nationale de France and the National Library of Medicine.

Implementation Specifics

The creation of Archival Resource Keys (ARKs) uses an identifier technology developed and operated by the California Digital Library. ARKs share a number of similarities with DOIs. These similarities include:

- Having both a URL form and a syntactic form independent of URLs
- Hierarchical decomposition
- Emphasis on opaque components
- A central, HTTP/URL-based resolver (http://n2t.net in the case of ARKs)
- The ability to associate metadata with identifiers.

ARKs are created through the aforementioned EZID tool, and in fact the interface for creating ARKs is identical to that for DOIs. Given the similarity between DOIs and ARKs, it may be natural to ask how one might choose between the two. Differences include:

- Infrastructure support: DOIs are based on the Handle system and administered by a distributed network of registrations agencies and allocators, so, enjoy broader infrastructural support. ARKs are currently supported solely by CDL (though efforts are underway to create a global ARK network).

- Cost: DOIs were formerly charged per identifier. That pricing policy has been dropped, but still, DOIs cost more than ARKs.

- Technical characteristics: the ARK specification supports partial resolution. The ARK resolver does not yet support this, but per CDL, implementation is forthcoming.

- Metadata: DOIs impose greater metadata requirements.

- Scope: DOIs created through DataCite are intended to identify citable data sets, i.e., data sets for which a journal-like citation can be formed. ARKs are unconstrained. As a result, ARKs may be more appropriate for identifying data set granules.

- Acceptance: Most publishers at this point may accept DOIs only; however, discussions with abstracting/indexing vendors to publish ARKs have commenced per ARK founders.

ARK Registry Services: Included in the ARK specification are generic service definitions for description, access, and location that specify the return of the object or a copy, a redirect to the same, or to a sensible object proxy such as a home page instead of an entire web site hierarchy. In addition, the specification states that if access is denied, an explanation of the object's current inaccessibility should be returned. The specification also defines a generic policy service for declarations of permanence, naming, etc., that should be returned for given ARKs. Object descriptions are returned in either a structured metadata format or a human readable text format; sometimes one format may serve both purposes. Policy sub-areas may be addressed in separate requests, but the following areas should be covered: object permanence, object naming, object fragment addressing, and operational service support. From our review, the EZID Service meets the ARK specification for registry services.

Persistent Uniform Resource Locator **(PURL)**

Relationship of PURL to URI/URL/URN

URI: A PURL is a URL, and so can be considered a URI. PURLs are usually assigned only to digital objects that are locatable on the WWW. Considered a URL whose task is to locate a resource, a PURL is an HTTP URI in a domain backed by a strong persistence policy.

URL: A PURL is a URL, but has no specific or proprietary syntax for the naming part of the URL.

URN: A PURL can become a URN by attaching the requisite URN: path prefix and adding a naming authority (/org/oclc) in front of the name (/purl/keith/home), as in URN:/org/oclc/purl/keith/home. By itself, however, PURL is not a registered IANA namespace**.**

Implementation Specifics:

The PURL system presents a central website for managing identifiers, domains, users, and groups. Establishing a user account was straightforward and free, and took a couple days to receive approval via email.

Once an account is created, it is necessary to create a "domain," i.e., the (unique) URL path component that follows the initial <http://purl.org/...> part of each PURL. We found this step to be surprisingly difficult. Our initial inclination was to use "esip" as the domain, since ESIP will be the owner and future maintainer of any identifiers we create. But embedding an organization name in an identifier is a risk to preservation, for if the ownership of the identified resource ever changes, the new owners may find it impossible or difficult or at the least awkward to maintain identifiers that carry another organization's imprint. While it is far better to use an opaque domain, it is oddly difficult for humans to manufacture

opaque strings (just as it is difficult for humans to generate random numbers). EZID provides an identifier "minting" capability for just this purpose, but PURL provides no such service. We ended up using a portion of the opaque identifier previously obtained from EZID as our domain.

Once a domain is created, creating identifiers with different types of redirects is straightforward. For example, we created the partial redirect <http://purl.org/5060D4/glacier_photos/>, with the effect that the PURL <http://purl.org/5060D4/glacier_photos/{photo_id}> will correctly resolve to any glacier photo so long as NSIDC maintains the practice of embedding the {photo_id} in the photo URL.

PURL also offers an API for batch operations, and we used this batch interface to create individual PURLs for each of the glacier photos. Unfortunately, the link to the API documentation on the PURL website is (ironically) broken, and other documentation we found was incorrect and appeared to be outdated. Only after considerable searching were we able to find some sample code that did work against the current PURL system. Because of the difficulty in reverse-engineering the batch interface, we have included some sample Python code in an appendix at the end of this report.

Finally, we note that PURL provides no means of associating metadata with an identifier, whether to support citation or for management purposes.

Universally Unique Identifier (UUID)

Relationship of UUID to URI/URL/URN

URI/URN: By virtue of its capability of serving as a URN, a UUID can be considered a URI.

URL: A UUID is not a URL by itself. At present, no mechanism exists to natively resolve the location of a UUID.

Implementation Specifics

We did not test UUID identifiers except in conjunction with the testing of Object Identifiers (OID) discussed below because there is neither centralized infrastructure support nor any services associated with UUIDs. As a result, any kind of functionality based on UUIDs (resolution, metadata, etc.) would have to be provided entirely by the user. We will note, however, that UUID support is included in all major programming languages, and such identifiers are easy to create, so they are most useful as unique identifiers.

Object Identifier (OID)

Relationship of OID to URI/URL/URN

URI: An OID is not a URI.

URL: An OID is not a URL and does not support use as a locator.

URN: An OID can be expressed as a URN by prepending "urn:oid:". It is an IANA registered URN namespace per RFC3061.

Implementation Specifics

Object identifiers (OIDs) are hierarchically organized, with each node in the hierarchy (i.e., each identifier) assuming complete and independent control over its sub-nodes (sub-identifiers). The root OIDs are defined by various standards. Significantly, there is no central or universal infrastructure supporting OID allocation, description, or maintenance. There is an OID Repository, but it is not an official registration authority of OIDs; it represents an attempt by an unnamed party to capture and describe existing OIDs. The OID Repository's disclaimer reads, "This OID repository is not an official Registration Authority, so any OID described in this OID repository has to be officially allocated by the Registration Authority of its parent OID. This OID repository gathers information about OIDs that has been submitted by any user of this web site." [10]

Given this decentralization, how does one actually register an OID? There are at least three routes. The first is to start a new standards track process, far too long and labor intensive a process for our purposes and, we suspect, for most scientists simply wanting an identifier for their dataset.

A second route is to register with IANA <http://www.iana.org/> a Private Enterprise Number (PEN) <http://pen.iana.org/pen/PenApplication.page> for an organization. While we made efforts to locate an existent PEN that could be used for purposes of the testbed from any ESIP member organization, we did not find a PEN that was being actively used. From this exploration, we came to believe that inserting an organization name into an identifier is a risk to preservation, since it can make maintenance of the identified resource difficult, if not impossible, should the resource's owner change. In the case of OIDs, this is true despite the use of opaque numeric components. Because of the fixed hierarchical nature of OIDs, any OID is forever owned and managed by the parent node, i.e., by the original organization that registered it.

We chose to test a third option, using the aforementioned OID Repository's capability of registering UUID-based OIDs under parent OID 2.25 <http://www.oid-info.com/get/2.25>. This produces uncomfortably long identifiers (e.g., 2.25.69932820419785521730996616709014930715), but ease of use is a virtue.

Such UUID OIDs are created by submitting a web form request to the OID Repository administrator that includes metadata such as a description of the identified resource and the requester's name and contact information. The administrator responded to our requests sometimes within a day and sometimes within a week. From the timing and nature of the responses, it was clear we were dealing with an individual human being. We also sent independent emails to the administrator, some of which were never answered.

[10] See http://www.oid-info.com/.

Interestingly, the administrator saw fit to edit our identifier requests. For example, a request for an identifier for "NSIDC Glacier Photo Collection" was approved but came back with the message "Note that the description has been changed to NSIDC. It is up to you to create a child OID for the gallery. The identifier has also been changed to 'nsidc'." Other submissions were denied entirely. A request for an identifier for "MODIS Snow Cover Climate Modeling Grid (CMG), Monthly Level 3 Global Product at 0.05Deg Resolution" (the exact title of a data product hosted by NSIDC) was denied with the explanation, "The description does not makes sense. Such an OID can only be assigned to a company or an international project. It is not clear what will be identified."

It appeared to us that the repository administrator would accept requests only for identifiers for organizations. As a final test, we tested whether we could follow this intention, and take advantage of the OID hierarchical model at the same time, by creating a single OID representing the NSIDC organization and then creating additional OIDs under that organizational OID for individual datasets hosted by NSIDC. In theory, as administrators of this newfound namespace we should have automatically received emails asking whether to accept the dataset registrations. However, we never received any such emails, and our sub-OID requests were denied by the repository administrator. Thus, at least as far as these UUID OIDs are concerned, the hierarchical model does not work.

Extensible Resource Identifiers (XRI)

Relationship of XRI to URI/URL/URN

URI: An XRI is transformed into a URI by adding "http://xri.net/" at the beginning and then appending the XRI. This URI then resolves to an XRDS (Extensible Resource Descriptor Sequence) document which is a simple XML document. XRIs can resolve to an XRDS document also by use of the HTTP(S) protocol. The second form is called HTTP XRI or HXRI. Other parameters can be added to the HXRI to further specify the resolution of the XRI. [11]

URL: By using the proxy resolvers described above, an XRI can resolve to the location of a digital resource.

URN: While XRI is not registered as a URN per the URN Syntax (RFC2141), a properly designed XRI can meet the requirements for RFC 1737 (Functional Requirements for Uniform Resource Names). XRIs were designed to offer additional features that URNs don't, i.e., support for XRI defined synonyms for a resource, global context symbols for identifier authorities, and a syntax for cross referencing other URIs. [12]

---

[11] See: https://www.fullxri.com/documentation/DocumentationPage/category/1/entry/4/ for documentation from one of the registry services about the advantages of XRI.

[12] See: http://equalsdrummond.name/2005/08/02/urns-and-open-tagging/ and http://en.wikipedia.org/wiki/Extensible_resource_identifier.

Implementation Specifics

We were unable to test the Extensible Resource Identifier (XRI) system. As noted on the XRI Wikipedia page, there has been contention over the system's adoption: "The XRI 2.0 specifications were rejected by OASIS, a failure attributed to the intervention of the W3C Technical Architecture Group which recommended against using XRIs or taking the XRI specifications forward." We observed no evidence of any significant activity on XRIs (for example, specification documents edited) since 2008.

The XRI resolver, http://xri.net/, currently redirects to http://inames.net/, the "i-name" resolver. i-names are one form of XRI and are intended to be a replacement for the DNS system. We looked for evidence of adoption of i-names, but found that major Internet corporations (e.g., Google, Facebook) have not registered i-names for their own domains.

Even if operational, i-names would seem to offer no advantage over DOIs. i-names are paired with i-numbers (opaque strings of hexadecimal digits). Because i-names may change, for persistence one must reference i-numbers. In addition, resolution of i-names requires embedding the i-name in a URL that references a central resolver.

Handles

Relationship of Handles to URI/URL/URN

URI: Handles can be used natively, or expressed as Uniform Resource Identifiers (URIs). Although the Handle System is not currently a registered stand-alone implementation of URI, it is a part of the info URI specification, RFC 4452.[13]

URL: Handles may also be expressed as Uniform Resource Locators (URLS), by the use of an http proxy server. [14]

URN: Handles can be used natively, or expressed as Uniform Resource Names (URNs) . Although the Handle System is not currently a registered stand-alone implementation of URN, it is a part of the info URI specification, RFC 4452. [15]

Implementation Specifics:

The Handle System is formally defined by a set of network protocols (RFCs 3650, 3651, and 3652), but these protocols are binary and complex and, as a result, non-trivial to implement. Thus it is safe to say that, in practice, the Handle System is defined by a Java software distribution provided by CNRI <http://www.handle.net/>. The system can be run locally to define local identifier namespaces and local resolution over those identifiers, but its relevance to persistent identifiers is its ability to define

[13] http://www.rfc-editor.org/rfc/rfc4452.txt

[14] http://www.handle.net/proxy.html

[15] http://www.rfc-editor.org/rfc/rfc4452.txt

global identifier namespaces. To operate globally, one must run a local Handle server, but "home" (register) the server with the global distributed Handle infrastructure. In short, identifiers are created locally, but are globally resolved by the CNRI's global server communicating with the local Handle server.

We created an Amazon EC2 virtual machine instance and installed the Handle software on the instance. Installation was straightforward. We did encounter some difficulties in firewall rules (Handle requires that a certain port be opened up for both TCP and UDP access), but strictly speaking such problems are not the fault of the Handle system, and to its credit, the Handle FAQ documentation describes the problem and possible solutions. The Handle system requires payment to participate in the global infrastructure, and it appears that payment is automated through PayPal (we communicated directly with CNRI to avoid paying for the purposes of this testbed).

The software distribution includes a Java client to communicate with the local server to create and manage identifiers. The client offers several modes of operation, including a GUI, a batch operation controlled through a GUI, and a command line operation. We tried all these and found them to work flawlessly.

The Handle system is capable of storing arbitrary metadata with identifiers, but beyond target URLs and some administrative metadata, this capability appears to be unused. We note that neither of the major DOI registration agencies (CrossRef and DataCite) store citation metadata in the Handle system. Thus, while it is possible to store metadata, there may be no broader advantages for doing so.

Life Science Identifiers (LSID)

Relationship of LSID to URI/URL/URN

URI: By virtue of being included in the URN namespace (and URN being is a URI), LSID is a URI.

URL: There are mechanisms for building a resolution service using URLs with LSIDs with an LSID Server Framework, but an LSID per se is not a URL.

URN: LSID is not a registered URN namespace per the IANA registry, but is considered a URN by using the form urn:lsid:authority:name:.

Implementation Specifics

An LSID qualifies a local identifier with a domain name and wraps the combination in URN syntax. For example, in the LSID urn:lsid:ncbi.nlm.nig.gov:GenBank:T48601:2, local identifier "GenBank:T48601:2" is qualified by the domain name of the organization to which the identifier belongs, "ncbi.nlm.nig.gov". There is no centralized infrastructure for LSID, nor is any required. Identifiers may be defined by any means (see the previous discussion under ARK for identifier generation strategies), and then simply prefixed with the organization's domain name. For this reason we did not test LSID.

The principal limitation of LSID is its dependence on domain names, which may change over the course of an object's lifetime. LSID is structurally similar to the "tag" URI scheme (RFC 4151), which provides a

mechanism for unambiguously referring to a domain name that may change ownership over time. LSID lacks both the capability for a domain to change ownership and for an identifier to change domains.

LSID developed a clever redirection mechanism by exploiting the DNS system, thus enabling LSID to serve as unique locators. It is not clear to what extent this mechanism has been adopted. In any case, LSID as an identifier scheme has generally not achieved adoption outside certain biological informatics communities. In the words of one researcher involved with LSID development, "The experiment [in LSIDs] has not gone well, in the sense that the support for LSIDs has waned over the years, and the only active use of LSIDs comes from a small community in biodiversity informatics. From my perspective, the LSID spec is all but abandoned."

Conclusions

Our task was to explore and report upon the operational issues associated with the implementation of each of the eight identifier schemes to two different types of canonical Earth Science data sets, and their components.

From an operational perspective we found DOI, ARK, and Handle identifiers to be suitable for Earth Science data sets and the most robust of the schemes tested in terms of ease of use, support, adoption, and scalability. DOIs and ARKs are perhaps easier to use than Handles because of the existence of centralized services (both human-oriented and APIs) for creating and managing identifiers; using Handle identifiers requires running a dedicated, local server.

UUID and LSID identifiers may also be suitable for Earth Science data sets. These identifier schemes are localized in nature; there is no central infrastructure as there is with DOIs, ARKs, and Handles. UUIDs would require significant infrastructural development to support anything other than the unique identification use case. LSIDs have gained little adoption since their initial release.

We found XRI and OID least suitable for use. XRI does not appear to be operational, and even if operational, would seem to offer little practical advantage over DOI. OID appears to be targeted at resource types other than data sets such as file formats and country names.[16]

While our conclusions mostly echo the descriptions and observations of the Duerr *et* al paper, they are based on experiential testing in a fairly realistic context, and on interviews with people who had real world experience implementing many of the identifier schemes. The value of our efforts and our conclusions lies in the empirical basis upon which they were developed.

[16] http://www.oid-info.com/faq.htm#2

Appendix A. Identifier Scheme Comparison Table

[See companion file in PDF form: AppendixA_IDComp.pdf]

Appendix B: Sample PURL batch code

Here is some Python code that creates a single PURL identifier using PURL's batch interface.

```
#! /usr/bin/python

# Based on:
# http://code.google.com/p/persistenturls/wiki/PURLClients

import cookielib
import urllib2

# Login and obtain authentication cookie.
opener = urllib2.build_opener(
  urllib2.HTTPCookieProcessor(cookielib.CookieJar()))
connection = opener.open(
  urllib2.Request("http://purl.org/admin/login/login-submit.bsh",
    data="id=username&passwd=password"))
connection.read()
connection.close()

# Now create a PURL.
request = urllib2.Request("http://purl.org/admin/purls")
request.add_header("Content-Type", "application/xml")
request.add_data("""<?xml version="1.0" encoding="ISO-8859-1"?>
<purls>
  <purl id="/domain/identifier" type="302">
    <maintainers>
      <uid>username</uid>
    </maintainers>
    <target url="target"/>
  </purl>
</purls>""")
connection = opener.open(request)
print connection.read()
connection.close()
```