Earth Sci Inform (2017) 10:383–393
DOI 10.1007/s12145-017-0297-3

METHODOLOGY ARTICLE

A service-based framework for the OAIS model for earth science data management

Edward Flathers[1] · **Jeremy Kenyon**[2] · **Paul E Gessler**[1]

Received: 18 August 2016 / Accepted: 25 January 2017 / Published online: 15 February 2017

Abstract The Consultative Committee for Space Data Systems (CCSDS), in 2002, released their first version of a Reference Model for an Open Archival Information System (OAIS). In 2003, the model was adopted by the International Standards Organization (ISO) as ISO 14721:2003. The CCSDS document was updated in 2012 with additional focus on verifying the authenticity of data and developing concepts of access rights and a security model. The OAIS model is the basis of research data management systems across institutions and disciplines around the world. The Organization for the Advancement of Structured Information Standards (OASIS), in 2006, released their first version of a Reference Model for Service Oriented Architecture (SOA). OASIS defines the SOA as "a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains." Systems designed around the SOA model benefit from improved scalability, flexibility, and agility. This paper applies the SOA model to the OAIS repository to describe how repositories can be implemented and extended through the use of services that may be internal or external to the host institution, including the consumption of network- or cloud-based services and resources. We use the Service Oriented Architecture (SOA) design paradigm to describe a set of potential extensions to OAIS Reference Model: purpose and justification for each extension, where and how each extension connects to the model, and an example of a specific service that meets the purpose.

Keywords Open archival information system (OAIS) · Repositories · Data management · Service oriented architecture (SOA)

Responsible editor: H. A. Babaie

✉ Edward Flathers
flathers@uidaho.edu

Jeremy Kenyon
jkenyon@uidaho.edu

Paul E Gessler
paulg@uidaho.edu

[1] College of Natural Resources, University of Idaho, Moscow, ID, USA

[2] University Libraries, University of Idaho, Moscow, ID, USA

Introduction

Responsibility for managing data created in a laboratory or via field work has traditionally been held by researchers. Over time, this has led to a great diversity of scientific data management practices differing in thoroughness of documentation, application of technology, and preservation of data (Tenopir et al. 2011). As our capacity to collect data increases with the proliferation of sensor networks and new instruments and simulation methods, we face a "data deluge" that easily overwhelms many of our traditional data management efforts (Hey and Trefethen 2003). A significant component of the data deluge, and one that generates a growing level of funding opportunities in data management research, is so-called "big data" (Haendel et al. 2012).

Big data is a term used to describe data that are usually characterized by the "three Vs" of volume, velocity, or variety (Zikopoulos et al. 2011). Data of high volume are large in size and can require large storage and network bandwidth resources to manage. For example, in 2012, the National Institutes of Health's 1000 Genomes Project exposed more than 260 terabytes of genetic data in more than 250,000 files (Clarke et al. 2012). Data of high velocity come into

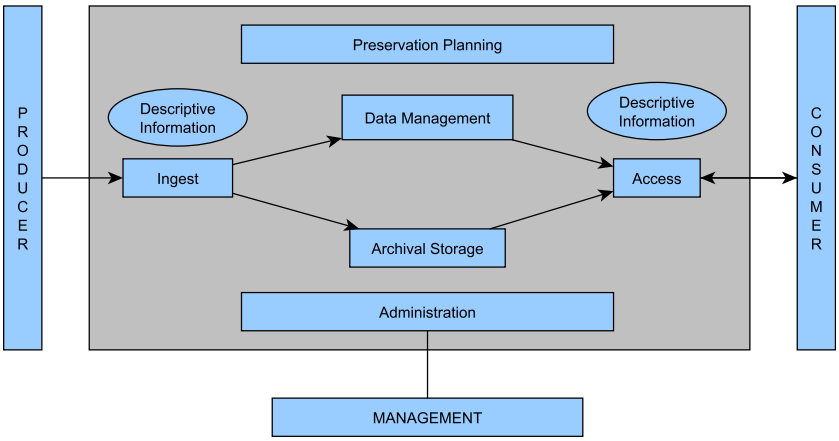

Fig. 1 The block diagram of the OAIS model (After CCSDS 2012 Fig. 4-1)

management systems, often from sensor systems, very quickly. The ATLAS Detector at the Large Hadron Collider creates 40,000,000 events per second, and filters out all but 200 per second, leaving them with a data recording rate of 320 megabytes per second (Haeberli et al. 2004). Data of high variety have inherent heterogeneity that can make them difficult to collate and compare. Take, for example, one minute's worth of social media postings—2.5 million Facebook posts; 300,000 Tweets; 220,000 Instagram photos; 72 h of YouTube videos—together, they tell a story about social media users, but each type of content requires a different set of tools for analysis (Gunelius 2014). Additional characteristics of big data have been identified in industry, but these three suffice to describe the challenges posed by big data in this paper.

One of the principal new challenges introduced by big data is data storage and curation (Hilbert and López 2011). As the information technology infrastructure needed to support research data grows in complexity and cost, the tasks of procurement and management can grow beyond the scope of the typical research project. Domain-specific and institutional data repositories have emerged to take advantage of economies of scale and provide standards-based methods for data storage and curation. To illustrate, over the past four years, the Registry of Research Data Repositories (re3data.org) has compiled a steadily growing registry of more than 1500 research data repositories in more than 60 countries.

The definition and design of repositories has been developing in parallel to the emergence of the repositories, themselves. The Consultative Committee for Space Data Systems (CCSDS), in 2002, released their first version of a Reference Model for an Open Archival Information System (OAIS) (Fig. 1). In 2003, the model was adopted by the International Standards Organization (ISO) as ISO 14721:2003. The CCSDS document was updated in 2012 with additional focus on verifying the authenticity of data and developing concepts of access rights and a security model. The OAIS model is a good fit for research data repositories, having been designed as a framework to support data collections without regard to data types, storage formats, access methods, or other specific implementation details. Among other agencies, the Library of Congress, NASA, the ESA, and the USGS apply the OAIS model for science data management.

The OAIS model involves bundling data and metadata into an Information Package that enables the basic functions of the repository: "ingestion, preservation, and dissemination of archived materials" (LaVoie 2014). There are four types of content contained within or associated with the Information Package: Content Information (CI), Preservation Description Information (PDI), Packaging Information, and a Package Description (PD) (Fig. 2).

The CI is made up of a Content Data Object—the data content of the package, and Representation Information (RI)—the metadata associated with the data content. Data content is often stored in a format compatible with the software used to record it, such as Microsoft Excel, ArcGIS, and other general or specialized programs. The metadata stored in the CI describes the data: data identification, contact information, collection methods, accuracy assessments, and others. In the Earth sciences context, metadata are often stored in standard formats such as the Federal Geographic Committee Content Standard for Digital Geospatial Metadata (FGDC CSDGM), the International Standards Organization's

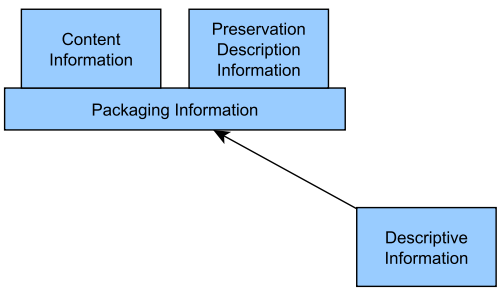

Fig. 2 Information package concepts and relationships (After CCSDS 2012 Fig. 2-3)

Geographic Information schema (ISO 19115), Ecological Markup Language (EML), and others (Goodchild 2007).

The PDI can be thought of as another set of metadata that is intended to describe information about the preservation and longevity of the CI. PDI describes five categories of information: Provenance, Context, Reference, Fixity and Access Rights (CCSDS 2012). In some cases, these categories may be described within the RI as well—for example, the ISO 19115 metadata standard defines elements for storing metadata within all five of the PDI categories. Regardless of metadata standard, however, some aspects of the PDI cannot be found within the RI because they are not determined until after the creation of the CI. For example, PDI can track information such as when the CI was added to the archive, what users of the archive have access rights to the package, and other facts relevant to the management needs of the archive. The operational details contained in PDI also help to verify the integrity of the archive, for example by storing checksums that alert administrators to changes in the contents of CI, enabling audits of the repository.

A working group co-sponsored by the Online Computer Library Center (OCLC) and the Research Libraries Group (RLG) developed the Preservation Metadata Implementation Strategies (PREMIS) metadata schema specifically for the purpose of implementing preservation metadata in the case of both RI and PDI (PREMIS 2008). The PREMIS metadata standard does not contain elements commonly used to describe geospatial datasets; this is an intentional limitation of scope by the standard's developers due to the existence of geospatial metadata standards listed above and others (PREMIS 2008). Therefore, the RI is better served using a domain-specific standard. However, it may be desirable to describe the PDI using PREMIS metadata, for example in order to standardize the structure of PDI in a repository with heterogeneous metadata using different standards.

The Packaging Information describes the organizational structure of the CI at the computer operating system level—file and directory structure that may be described by ZIP (formalized as ISO/IEC 21320–1:2015) or BagIt (Kunze et al. 2011) archives, or other aggregation schemes. The PD contains information used by data consumers to search for and retrieve the complete Information Package, such as title and abstract fields. The PD can be extracted from the RI and the PDI and inserted into an index to support search and browse functionality.

As the implementation details of the OAIS model are intentionally omitted from the specification, the software design for the repository itself, as well as for related functions, is left as a choice to the architects of such systems. Here, we advocate for the Service-Oriented Architecture (SOA) as an ideal approach to implementation. According to the Reference Model for Service Oriented Architecture developed by the Organization for the Advancement of Structured Information Standards (OASIS), SOA is "a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains" (OASIS 2006). SOA is a concept from computer sciences that describes building modular, loosely-coupled software systems (Papazoglou and Van Den Heuvel 2006). The modules, or services, that are deployed in such a system may exist in geographically disparate locales; they may be created and maintained by separate institutions or groups, and they may rely on entirely different computing hardware and software. The loose mode of coupling is accomplished through the exposure of an Application Programming Interface (API) that explicitly defines the language and communication protocol through which the service interacts with the outside world. As long as a service properly implements the requirements of the API, it can interoperate with other systems that speak its language. This is opposed to the concept of a "tightly coupled" system, in which components may communicate with each other through channels and protocols that are opaque to outside observers and are generally meant to be invoked only from components within the system, itself.

There are a variety of general motivations for implementing complex software systems using SOA:

- SOA can enhance system reliability: because the system is composed of multiple modules, the failure of any one module does not necessarily mean the failure of the entire repository function, whereas the failure mode of monolithic software systems may bring down the entire suite of functionality (Tsai 2005).
- SOA enables staggered rollout of new features: since service modules (outside a core set of modules) are independent of each other, new features can be implemented as the repository is operating and introduced publicly when they are ready for consumption (Wong-Bushby et al. 2006). In this fashion, an SOA-based OAIS repository can be 'bootstrapped' into a full-featured state over time.
- SOA preserves the functionality of legacy systems: based on the loosely-coupled philosophy of SOA, implementers can design linkages between legacy systems such as institutional/enterprise management software and repositories (Pessoa et al. 2008). As legacy systems transition to more modern versions, linkages can be adjusted to compensate for varying modes of interaction.
- SOA supports interoperability with external systems: similarly to the linkage to legacy systems, loose coupling also supports linkage to systems that exist outside the repository or the institution (Nezhad et al. 2006). Modern systems that are designed for interoperability use standard or well-known APIs that lessen the effort involved in connecting to them from remote systems. Furthermore, repositories designed with interoperability in mind enable catalog and data consumption from external services using standard APIs.

- SOA improves upon the flexibility of monolithic software: one of the challenges of deploying monolithic software solutions is that they are typically designed for a use case that is not exactly reflected within the institution. There may be a component that is missing or unsuited to the environment in which the system is to be deployed. The modular approach of SOA allows implementers to choose individual components from available options, or to implement a particular component themselves (Ren and Lyytinen 2008).
- SOA separates development into manageable tasks: because the modules of an SOA-based repository take advantage of loose coupling and APIs to interact with each other, maintenance, bug fixes, and development work done on one component do not immediately require making changes to the internal code of another. If new functionality is required of the repository, the functionality can be implemented one component at a time, reducing the complexity of development tasks. When APIs are updated with new functionality, older functionality can be maintained by continuing to support older versions of the API (Josuttis 2007). This can help to maintain links to legacy and external systems.
- SOA allows distribution of repository functions across geography and institutions: as interdisciplinary research and large-scale collaboration increase in popularity, it is important that data management systems are able to federate functionality and content with each other (Yarmey and Khalsa 2014). Even standards-based repositories do not always follow the same standards, especially across international borders. The SOA approach to interoperating with external systems can be crucial for communication across institutions.
- SOA allows the compartmentalization of user access rights and security (Channabasavaiah et al. 2003). Since each service operates using its own security model and user authentication requirements, privileged access can be reserved for users and modules that definitely require heightened levels of access.
- SOA helps to avoid problems associated with vendor lock-in (Brown et al. 1998). With monolithic software, administrators face deadlines such as end-of-life dates, at which the entire software package must be upgraded to a newer version, regardless of whether the newer version represents an improvement over the old one for users.

A theme that emerges among the strengths of SOA is ease of adapting to change. In order to provide value, the continuing development of science data repositories must be driven by the dynamic needs of the research communities that feed them. The data deluge involves research products that are growing in size and complexity faster than existing systems can accommodate (Hey and Trefethen 2003). Repositories must be prepared to adapt to support data of various scales, from small legacy text-based data to newer terabyte- or higher-scale collections. New science and technologies often involve data stored in novel file or database formats (Ahrens et al. 2011). As these novel formats proliferate, they enable an increasingly heterogeneous list of new features and capabilities, pushing repositories to expose new and updated services. As repositories are driven to federating and other methods of interoperability, they must adapt to the choices and limitations of technologies implemented by potential partners. These and other adaptations are strongly supported by the SOA approach.

There are also limitations to the SOA approach to developing repositories. The need to adapt SOA-based repositories to accommodate new conditions represents engineering challenges for software developers (Palma et al. 2013). Keeping the various services of the system functioning and interoperating smoothly can be another challenge. Relying on monolithic software allows repository administrators to focus on the business of managing and curating data, rather than overseeing the continued development and maintenance of software services.

There are a variety of monolithic, off-the-shelf software choices for repositories. According to the Registry of Research Data Repositories, which surveys research data repositories worldwide, out of 1763 repositories, the top three data management systems are DSpace (42 instances), DataVerse (36 instances), and CKAN (28 instances) (Re3Data 2016). These numbers likely underestimate the number of repositories using these software packages—the vast majority (1266 instances) are listed as either “other” or “unknown”. Amorim et al. (2016) presents a more complete list of repositories and performs some evaluation of their relative merits. Some, like DSpace, specifically aim to implement the OAIS model, but most do not.

Adherence to the OAIS model for repositories comes with several advantages. First, OAIS-based repositories take advantage of the deep thought and planning by a large body of researchers that has gone in to building the model. Second, the CCSDS is developing a recommended practice for the Audit and Certification of Trustworthy Digital Repositories “to create an overall climate of trust about the prospects of preserving digital information” (CCSDS 2011). Furthermore, the application of a standard model may serve to improve interoperability between repositories due to the use of common paradigms in design and implementation.

The OAIS model explicitly “does not specify a design or an implementation” (CCSDS 2012). In part due to this, and also due in part to the uncertain speed, reliability, and persistence of Internet connections during the inception of the OAIS model, one area that is not well-developed is the connection of data repositories to network- or cloud-based services and resources. We use the Service Oriented Architecture (SOA)

design paradigm to describe a set of extensions to the OAIS Reference Model that enable a repository to take advantage of recent opportunities for interoperability. We describe a purpose and justification for each extension, where and how each extension connects to the model, an example of a specific implementation that meets the purpose, and a suitable API definition to support the functional purpose.

Methods

Data unique identifiers

In order for data consumers to make use of data in repositories, the data must have a persistent point of access and must be verifiably the data that the consumer is interested in. Unique identifiers for data can provide access keys that decouple location information from identifiers so that when data are moved, identifiers remain consistent while location information is updated in linked databases. By maintaining a consistent identifier for data, citations in publications and other documents resist becoming stale, so consumers can maintain access to data and be sure they are accessing the data they are expecting.

In 2011, Duerr et al. published an assessment of nine different data identification schemes: ARKs, DOIs, XRIs, Handles, LSIDs, OIDs, PURLs, URIs/URNs/URLs, and UUIDs. Of these, the Digital Object Identifier (DOI) stands out as a strong candidate for application in data repositories given its widespread adoption by publishers, its acceptance as an ISO standard (ISO 26324), and its interoperability with other common location and identification schemes such as Uniform Resource Identifiers (URIs). The DOI is a product of the International DOI Foundation "designed as a generic framework applicable to any digital object, providing a structured, extensible means of identification, description and resolution (International DOI Foundation 2012)."

The DOI works via a central registry that associates unique identifiers with the locations of data products. The recommended practice for assigning the endpoint of a DOI is not to link directly to data products, but to web pages that display descriptive information about the data products, often to include download links or instructions for obtaining the data if not available for download (International DOI Foundation 2012). This descriptive information can be derived directly from a repository's representation information, providing the consumer with some certainty that they have found what they are looking for.

The infrastructure requirements for accommodating the DOI are modest. It requires a method of storing the DOI value such that it is associated with the data object that it identifies, a method of discovering DOI values that are stored within the repository, and a method of resolving client requests for DOIs.

The ideal storage location of the DOI is within the metadata associated with a data object, as this identification information is solidly within the purview of the purpose of Representation Information. However, not all metadata standards allow for the storage of a DOI in an unambiguous way. The FGDC CSDGM, for example, defines no specific location for the storage of a DOI. Although one could be stored in a variety of locations within a metadata satisfying the standard, the weak semantic cues given by more general-purpose fields makes it difficult for an automated process to identify unambiguously that a DOI that it finds within them is the correct identifier for the dataset. To account for cases in which the representation information standard does not allow for unambiguous storage of the DOI, the PDI can also be used to store the DOI using a standard such as PREMIS or an ad-hoc approach.

The discovery method for the DOI can be as simple as for any other field within the metadata: index the DOI field and present it through the normal search interface.

The data unique identifier module integrates with the OAIS model in three places: at the ingestion phase, where a user can input or assign the DOI information relevant to the record being inserted; in the storage system, where the Packaging Information associates the DOI with the repository record; and at the access phase, where users can query the repository based upon the DOI.

Some issuers of DOIs, such as the California Digital Library's (CDL) EZID service (http://ezid.cdlib.org/), expose an API that allows clients to request a DOI for a data object. At the ingestion phase then, the repository accesses the remote API to issue a new DOI for the ingested data object and inserts the DOI into the Representation or Preservation Description Information for storage.

Resolving client requests for access to data identified by DOIs involves accepting a query for a DOI; looking up the DOI in the repository; and either retrieving and rendering a search result, or indicating the failure of the repository to resolve the DOI. Following the best practice for DOIs resolving to descriptive landing pages, the repository may generate a page upon request, based upon information from the RI and PDI. The dynamic generation of a landing page based upon the object's metadata helps ensure that landing pages always include the most up-to-date, authoritative description available for the Information Package (Fig. 3).

Researcher unique identifiers

One common difficulty in the academic publishing arena is the potential for ambiguity of authors' names. A researcher may, over the course of a career, publish under more than one name, making it difficult to assemble an exhaustive list of their publications. Multiple researchers may share the same name (or initials), making it difficult to separate their individual bodies

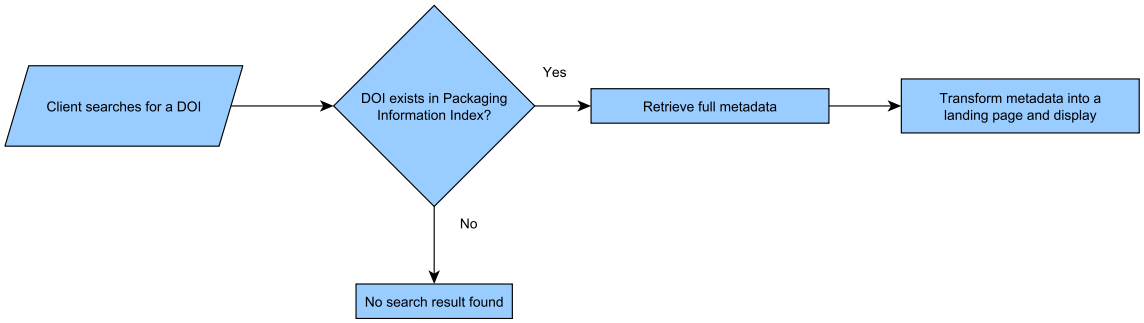

Fig. 3 An example of a Consumer interacting with the repository by requesting details of a DOI

of work. The combination of a researcher name and institution can help, but is still problematic when researchers change employers, have multiple appointments, or have common names and are associated with large institutions (Han et al. 2004).

One approach to disambiguating author names is to associate unique identifiers with authors. This approach requires a certain amount of cooperation between authors, who must agree to participate in a registry and keep their record up-to-date; publishers, who must agree to include the unique identifiers with publications; and a central authority that maintains the registry of mappings between authors and their unique identifiers. One such organization that has seen widespread adoption is Open Researcher and Contributor ID (ORCID), which operates a web-based registry (Haak et al. 2012).

The author identifier service has three useful points of interaction with the OAIS research data repository. The first is when the data producer initiates the ingestion process. The producer will be requested to create a certain amount of metadata to describe the data that they are submitting to the archive. As part of that metadata collection effort, the producer should be given the opportunity to provide their unique identifier.

As with the DOI data identifiers, most representation information standards have no explicit way to store ORCID or other systematized researcher IDs. The ORCIDs may be stored in a variety of ways within metadata, but are difficult to store in a semantically unambiguous way.

The ORCID API allows systems to query the ORCID database to retrieve public data about authors who are indexed in the system. A researcher unique identifier module, then, can interface with the OAIS repository in three ways.

The second point at which the author identifier can interact with the repository is during the ingestion process, when the ORCID is collected from the producer. The repository can use the ORCID API to query for and populate fields related to producer identification using the data that are publicly available from the ORCID database. This step can save time and effort for the producer by obviating the need to manually enter simple identification information.

At the storage phase, the repository then stores the ORCID in a designated field within the Packaging Information associated with the data object. As a part of periodic metadata maintenance, it is then possible to compare the stored ORCID for a data object with the producer information stored within the representation information and check for mismatches. It is not clear in these audits whether the metadata has fallen out of sync with the reality that is represented in ORCID or the other way around (alternatively, both the representation information and ORCID database may have become obsolete), but it is at least possible to use the audit to flag the record for a human to review and try to find a resolution.

The third point of interaction between the author identifier and the repository is at the data consumer interface, when a potential consumer wishes to search for data produced by a particular researcher. If the consumer is able to search using the producer's unique identifier as a key, the results that they retrieve should be unambiguous. As with the DOI, the discovery method for the ORCID can be as simple as for any other field within the metadata: index the ORCID field and present it through the normal search interface.

Federated user credential and identity management

With today's focus on interdisciplinary research projects that can span multiple institutions, researchers can face challenges in dealing with disparate information technology systems. One such challenge is user credential management—while each participant in a research project has a set of computer credentials issued by their institution, these credentials are rarely interoperable. That is, computer users at one institution cannot use their login credentials with systems at another institution.

These incompatible credentials lead to problems with the research data repository. The repository may support a standalone credential system, but how can repository operators know whether users from other institutions are allowed certain types of access? Even if users have authenticated with their home institutions, how can these credentials be trusted? On-line identity theft is a growing problem in the business sector, and can easily transition to the research world. Some research data may be protected by statutes such as FERPA or HIPAA, some may be protected by agreement with an institutional review board, and some may be sensitive due to their unique nature; it is therefore important to maintain a system of

credential management for repository users in order to control access and management of data assets.

A potential solution to this issue is federated credential management, a system in which institutions join together to vouch for the validity of their users' login credentials (Bhatti et al. 2007). There exist a variety of organizations providing federated credential services, many focused on particular geographic areas or activity domains. A popular provider among academic institutions in the United States is InCommon (Barnett et al. 2011). These organizations allow credential providers to issue usernames and passwords to their users and to share their authentication process with external systems without transmitting or revealing the actual credentials. In this way, individual institutions can continue to manage the basic details of user credentials such as login names and passwords while enforcing their own local policies. Federated credentials can interact productively with researcher unique identifiers as well: if credential stores contain ORCID information, and expose that information to systems consuming their authentication services, then federations can share not only credentials, but also identities.

From the repository perspective, managers can grant access rights to users based upon information gleaned from the federated credential service. Based upon common identifiers such as ORCID, repository managers can arrange permissions to allow individual users or groups of users to create, modify, or view data packages stored in the repository.

The federated identity system can connect with the OAIS model at any point of connection into the archive from outside: producer, consumer, or manager. The mode of connection is through the API exposed by the identity management system. This API is responsible for accepting authentication credentials and returning some base level of information about the user that has successfully logged in: at the minimum, a user ID that is compatible with the local repository system. Ideally, more information would be shared: user data such as ORCID and other descriptive data that help the repository to categorize the external user. Once a user has authenticated, access rights can be managed just as with any traditional, locally existing user. In this way, multiple repositories can share user identities without the need of sharing user credentials, and can grant privileges within the repository to users of other repositories to support collaboration across institutions. When identity information is included in addition to credentials, external users gain the benefits provided by the repository's researcher unique identifier module.

Harvesting, federated catalogs, and search

Given the proliferation of research data repositories—Marcial and Hemminger (2010) identified thousands of science data repositories in a survey in 2010—potential data consumers may not be aware of repositories that could hold information that would further their research goals. Rather than searching many repositories individually, it can be helpful to the user to be able to search many repositories simultaneously.

Two related approaches to expanding search capabilities to the content of multiple repositories are harvesting and federated search. Harvesting is a process of collecting remote metadata records into the local repository, ingesting them automatically for search. Federated search involves applying search terms not only to the local repository, but also to remote repositories to find results. For both of these approaches, a repository must provide a means of both supplying and consuming these services.

One popular way of arranging harvesting services is through the standard protocol, Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH). In order to harvest data from a repository that implements OAI-PMH, a harvester makes a "ListRecords" request and the remote repository responds with an OAI-PMH envelope that contains a series of records that list an identifier, a datestamp, and a metadata record conforming to a format specified in the request.

The first two of these items should be readily available from the packaging and representation information. The metadata may be more difficult to come by if the requested format is not the native format of the representation information. The bulk of the response implementation, then, is implementing some translation service that can produce at least a minimal metadata record based upon information found in the representation information. Since this metadata will only be used for data discovery purposes, only a small number of fields must be populated; the difficulty may arise from diversity of source locations for the content of these fields based upon the variety of representation information standards that are stored within the repository.

Federated Search is one approach for addressing this need. Federated Search provides the user with one search interface that connects with many back-end repositories and provides results in aggregate form (Shokouhi 2011). Federated Search is most easily accomplished when repositories offer a common search API, obviating the need for custom computer code to connect to different repositories. One common federated search protocol is the Open Geospatial Consortium's Catalog Service for the Web (OGC CSW) (Liakos et al. 2015).

Within the OAIS model, the standard search API would be implemented at the Access block that interfaces with the data consumer. Multiple access methods may be implemented, and there are several data repository systems that support multiple standards and ad-hoc methods of access.

At the heart of supporting federated search is exposing some amount of the packaging and representation information to other systems using a well-known API. Regardless of the metadata standards and implementations used within the repository, if the necessary details of data objects can be organized into valid API responses, then the data can be made searchable through federation.

Data object replication

Another set of challenges for potential data consumers can be dealing with slow transfer speeds resulting from long geographic (or network topological) distance to the repository, and data that are inaccessible due to repository or network down time. If users are unable to achieve reliable access to archived data, they are unlikely to rely on such data for their own research purposes.

One method for mitigating the risks of low-availability data is to replicate the data in multiple disparate geographic areas, decentralizing risk across networks and nodes. This can be done at several conceptual levels within the repository architecture. For example, the "Archival Storage" that the data consumer accesses may not be a single storage system, but a distributed file system such as can be accessed through the Amazon Simple Storage Service (S3) (https://aws.amazon.com/s3/). In a replication system implemented at that level, the repository itself need not be aware of the particulars of the geographic locations of files; the file system is abstracted sufficiently from the repository that at any geographic (or network-topological) location, a data consumer who accesses a data object is automatically given access to the nearest copy.

One approach to this replication is the NSF-funded DataONE project, which uses an OAIS-like implementation to ingest data into member nodes and then distribute replicas of data objects to several other member nodes in other locations around the network (Reichman et al. 2011).

In this mode of replication, the repository may need to be more involved. Data object replication can be thought of as a scenario in which an agent consumes data objects from one repository and produces those same objects for ingestion into a second repository. In order for data replication to occur in an automated and predictable way, a common data access API can be implemented at the Access block that interfaces with the data consumer. A data ingestion API can be implemented at the Ingest block of the model that interfaces with the data producer. In this case, a software agent interfaces with these APIs to connect two repositories. Such an agent may be operated by one or the other (or both) of the endpoints of the replication transaction and may require some supporting Packaging Information to be associated with the data objects, for example to indicate that a particular data object is an ideal candidate for replication.

A further benefit of replication is that it provides some redundancy of data object storage that can make data more robust against catastrophic events. Should one repository be struck by an irrecoverable data loss scenario, data that have been replicated to other sites should still be available. Though replication is not equivalent to, and should not be used in lieu of, a traditional backup system, it may serve a similar purpose.

Version control

As time passes, information contained in metadata tends to fall out of date, particularly in the case of information about people and institutions associated with data—names, phone numbers, addresses, the organization of institutions. These details tend to change over time. Much more rarely, changes will need to be made to the sections of metadata referring to the data, themselves. In either case, as changes are made to metadata records, it can be difficult to compare two metadata records and determine whether or not they describe the same dataset and are, in fact, two different versions of the same metadata record.

The issue of data provenance is important when considering using research data secondarily. It is critical that a researcher knows if changes have been made to a data object since its creator first published it into an archive, both to determine the data's suitability for use and to be able to accurately represent the full extent of data processing methods that have been applied.

Version control systems (VCS) offer the capacity to look back at previous versions of files that are stored within them and see in precise detail how those files have changed over time (Sen 2004). There are several popular version control systems today; foremost among them are Git (https://git-scm.com/) and Subversion (https://subversion.apache.org/). Version control can be deployed within an OAIS compliant repository's Archival Storage system.

For example, a VCS such as Subversion can operate in a way that is mostly transparent to the repository except when its special functions are needed. The repository continues to keep metadata in its usual storage system, registering each record with the VCS. As metadata records are updated by producers, the VCS keeps a history of each record, tracking changes to the metadata. Consumer users of the repository are presented with the latest version of a metadata record by default, but on request, the VCS can provide a detailed revision history. For consumers who are interested in previous versions of metadata records, there are many existing tools that allow powerful browse and search capabilities, such as the open-source Windows application, TortoiseSVN (https://tortoisesvn.net/). Using such tools, a data consumer can check previously downloaded representation information against old versions stored within the VCS to verify that they are using an older version of the same data object.

For repository administrators, the version control system provides an audit trail that allows them to identify who has made changes to a file, at what time, and of what substance. This information can be used in the development of detailed provenance records for data and metadata. Reporting on update activity can also give administrators insight into how data producers are interacting with the repository, which metadata records undergo frequent update, and why—potentially

helping to inform the kinds of training and assistance offered to producers. The VCS also grants administrators the capability of inspecting and reverting changes that have been applied erroneously as metadata records are maintained. Like data replication, VCS can offer a kind of backup capability for the repository, allowing damage to be undone

Taxonomies and controlled vocabularies

Taxonomy services provide access to controlled vocabularies for use by organizations and disciplines to classify things (Cohen 2007). In data management, the controlled vocabulary can be used to provide a consistent set of descriptive terms used to describe a dataset. Consistency enhances the ability of search clients to be able to locate records described by a particular term. For example, when using keywords to describe geographic data collected within the United States of America, it is useful to have a common term such as "USA" rather than a proliferation of variations such as "U.S.A.", "US", "U.S.", "United States", "America", etc.

A wide variety of taxonomy services exist, particularly services suited to certain research domains. The ISO 19115 Topic Categories is a simple example of a taxonomy intended to describe a general theme of geospatial data. It contains only 19 terms: farming, biota, boundaries, climatologyMeteorologyAtmosphere, economy, elevation, environment, geoscientificInformation, health, imageryBase MapsEarthCover, intelligenceMilitary, inlandWaters, location, oceans, planningCadastre, society, structure, transportation, and utilitiesCommunication (ISO 2007). The generality and limited number of terms of this taxonomy limit the ability to express complex information about a dataset, but do provide a standard set of terms that may be used in data discovery.

The USGS Geographic Names Information System (GNIS) (http://geonames.usgs.gov/) is a much more elaborate taxonomy that records the variety of official names for geographic features across the United States. Containing more than two million entries, this taxonomy can be used to specifically identify a geographic location, but can prove daunting as a search tool due to its size.

Taxonomy services are useful at two stages in the data ingestion process. First, the data producer can take advantage of the service while producing the metadata record. This application is beyond the scope of the data repository itself, but metadata creation/editing utilities may be designed to interface directly with the repository, so it can be of benefit to coordinate between any utilities created and any repositories used to ensure that they use common taxonomy services.

The second application of the taxonomy service occurs in the Quality Assurance block of the Ingest system. If the repository mandates the use of certain taxonomies where applicable in metadata, then the QA process can use the taxonomy service to verify the content of the relevant metadata elements, rejecting non-complying metadata for further review by producers.

Live data exposure

As the resolution of measurements across many dimensions increases with access to advanced instruments and massive storage systems, data consumers may prefer not to copy entire data sets for local use, instead opting to extract useful subsets or aggregations of data, or simply to connect to services that expose data and perform analyses remotely. When dealing with very large data collections, it makes sense to transfer only those parts of the data that are involved in analysis in order to conserve transfer time, local storage, and computational resources used in analysis.

To that end, data services such as the OGC Web Feature Service (http://www.opengeospatial.org/standards/wfs), the Unidata Thematic Realtime Environmental Distributed Data Services (THREDDS) Data Server (http://www.unidata.ucar.edu/software/thredds/current/tds/), the Consortium of Universities for the Advancement of Hydrologic Science Hydrologic Information Service (CUAHSI HIS) (http://his.cuahsi.org/), and others have arisen. These services provide a consumer-facing API that accesses the Archival Storage block to manipulate and expose data in formats that are friendly to the consumer's data client software, where they may then be analyzed and visualized as if they were local resources.

For many of these services to function, most of the repository system need not be involved directly. As an example, the THREDDS service can be set up as a new Access component to the repository; THREDDS becomes a new "Live Access" component of the repository, another way for the consumer to access the data.

Conclusion

The OAIS reference model is intentionally devoid of implementation detail, but our current climate of cloud- and network-based services lends itself to low-level interaction with some internal parts of an OAIS repository. We have described unique identifiers for data that help to provide long-term access and assure the identity of the data object; unique identifiers for researchers that disambiguate data producers and can help to identify a researcher's body of work; federated user identity management that provides a single set of credentials that enable access controls; federated catalogs and search that help make data objects accessible through more interfaces and to more potential consumers; data replication that can provide redundancy protection against certain kinds of data disasters and enables fast access to data objects by consumers; version control that provides audit

histories of metadata that allow for the comparison of metadata records; taxonomy services that help to control search vocabularies to help consumers search for data; and live data services that can obviate the need for data consumers to download large data objects in situations where they may only need small parts. Together, these services serve as a set of implementation details for an OAIS repository that are relevant to our modern level of connectivity and collaborative research.

Acknowledgements This material is based upon work that is supported by the National Institute of Food and Agriculture, U.S. Department of Agriculture, under award number 2011-68002-30191.

References

Ahrens J, Hendrickson B, Long G et al (2011) Data-intensive science in the US DOE: case studies and future challenges. Comput Sci Eng 13(6):14–24

Amorim RC, Castro JA, da Silva JR, Ribeiro C (2016) A comparison of research data management platforms: architecture, flexible metadata and interoperability. Univ Access Inf Soc:1–12. doi:10.1007/s10209-016-0475-y

Barnett W, Stewart CA, Walsh A, Welch V (2011) A roadmap for using NSF cyberinfrastructure with InCommon. http://hdl.handle.net/2022/13024 and http://www.incommon.org/nsfroadmap.html. Accessed 13 December 2016. doi:2022/13024

Bhatti R, Bertino E, Ghafoor A (2007) Federated identity and privilege management. Commun ACM 50(2):81–88. doi:10.1145/1216016.1216025

Brown WJ, Malveau RC, McCormick HW III et al (1998) AntiPatterns: refactoring software, architectures, and projects in crisis. John Wiley & Sons, New York

CCSDS: Consultative Committee for Space Data Systems (2011) Audit and certification of trustworthy repositories https://public.ccsds.org/pubs/652x0m1.pdf. Accessed 13 December 2016

CCSDS: Consultative Committee for Space Data Systems (2012) reference model for an Open Archival Information System (OAIS). https://public.ccsds.org/pubs/650x0m2.pdf. Accessed 13 December 2016

Channabasavaiah K, Holley K, Tuggle E (2003) Migrating to a service-oriented architecture, Part 1. https://www.ibm.com/developerworks/library/ws-migratesoa/. Accessed 01 Feb 2017

Clarke L, Zheng-Bradley X, Smith R et al (2012) The 1000 genomes project: data management and community access. Nat Methods 9(5):459–462. doi:10.1038/nmeth.1974

Cohen S (2007) Ontology and taxonomy of services in a service-oriented architecture. Microsoft Architecture J 11:30–35

Duerr RE, Downs RR, Tilmes C et al (2011) On the utility of identification schemes for digital earth science data: an assessment and recommendations. Earth Sci Inf 4:139. doi:10.1007/s12145-011-0083-6

Goodchild MF (2007) Beyond metadata: Towards user-centric description of data quality. Keynote paper, Proceedings, 5th Int. Symposium Spatial Data Quality, ITC, Netherlands, 13–15 June

Gunelius S (2014) The Data Explosion in 2014 Minute by Minute http://aci.info/2014/07/12/the-data-explosion-in-2014-minute-by-minute-infographic. Accessed 13 December 2016

Haak LL, Fenner M, Paglione L et al (2012) ORCID: a system to uniquely identify researchers. Learned Publishing 25(4):259–264. doi:10.1087/20120404

Haeberli C, dos Anjos A, Becket HP et al (2004) ATLAS TDAQ data collection software. IEEE Trans Nucl Sci 51(3):585–590

Haendel MA, Vasilevsky NA, Wirz JA (2012) Dealing with data: a case study on information and data management literacy. PLoS Biol 10(5):e1001339. doi:10.1371/journal.pbio.1001339

Han H, Giles L, Zha H et al. (2004) Two supervised learning approaches for name disambiguation in author citations. Proceedings of the 2004 joint ACM/IEEE conference on Digital Libraries, pp 296–305

Hey AJG, Trefethen AE (2003) The data deluge: an e-science perspective. In: Berman F, Fix GC, Hey AJG (eds) Grid computing: making the global infrastructure a reality. Wiley, New York, pp 809–824

Hilbert M, López P (2011) The world's technological capacity to store, communicate, and compute information. Science 332(6025):60–65. doi:10.1126/science.1200970

International DOI Foundation (2012) DOI Handbook http://www.doi.org/hb.html. Accessed 13 December 2016

ISO: International Organization for Standardization (2007) ISO/TS 19139:2007: Geographic information–Metadata–XML schema implementation http://www.iso.org/iso/catalogue_detail.htm?csnumber=32557. Accessed 13 December 2016

Josuttis NM (2007) Versioning. In: St. Laurent S (ed) SOA in practice: the art of distributed system design. O'Reilly Media, Sebastopol CA, pp 145–157

Kunze J, Boyko A, Littman J et al. (2011) The bagit file packaging format (v0. 97). https://tools.ietf.org/html/draft-kunze-bagit-08. Accessed 13 December 2016

Lavoie BF (2014) The Open Archival Information System (OAIS) reference model: introductory guide (2nd Edition) www.dpconline.org/component/docman/doc_download/1359-dpctw14-02. Accessed 13 December 2016

Liakos P, Koltsida P, Kakaletris G et al (2015) A distributed infrastructure for earth-science big data retrieval. Int J Coop Inf Syst 24(02):1550002. doi:10.1142/S0218843015500021

Marcial LH, Hemminger BM (2010) Scientific data repositories on the web: an initial survey. J Am Soc Inf Sci Technol 61(10):2029–2048. doi:10.1002/asi.21339

Nezhad HRM, Benatallah B, Casati F, Toumani F (2006) Web services interoperability specifications. Computer 39(5):24–32

OASIS: Organization for the Advancement of Structured Information Standards (2006) Reference model for service oriented architecture version 1.0. http://docs.oasis-open.org/soa-rm/soa-ra/v1.0/cs01/soa-ra-v1.0-cs01.html. Accessed 13 December 2016

Palma F, Nayrolles M, Moha N et al (2013) SOA antipatterns: an approach for their specification and detection. Int J Coop Inf Syst 22(04):1341004

Papazoglou MP, Van Den Heuvel WJ (2006) Service-oriented design and development methodology. Int J Web Eng Technol 2(4):412–442

Pessoa RM, Silva E, van Sinderen M et al. (2008) Enterprise interoperability with SOA: a survey of service composition approaches. 2008 12th Enterprise Distributed Object Computing Conference Workshops, pp 238–251

PREMIS: PREMIS Editorial Committee (2008) PREMIS data dictionary for preservation metadata version 2.0. http://www.loc.gov/standards/premis/v2/premis-2-0.pdf. Accessed 13 December 2016

Re3data: Registry of Research Data Repositories (2016). http://www.re3data.org/. Accessed 11 August 2016

Reichman OJ, Jones MB, Schildhauer MP (2011) Challenges and opportunities of open data in ecology. Science 331(6018):703–705. doi:10.1126/science.1197962

Ren M, Lyytinen KJ (2008) Building enterprise architecture agility and sustenance with SOA. Commun Assoc Inf Syst 22(1):4

Sen A (2004) Metadata management: past, present and future. Decis Support Syst 37(1):151–173. doi:10.1016/S0167-9236(02)00208-7

Shokouhi M (2011) Federated search. Found Trends Inf Retr 5(1):1–102. doi:10.1561/1500000010

Tenopir C, Allard S, Douglass K et al (2011) Data sharing by scientists: practices and perceptions. PLoS One 6(6):e21101. doi:10.1371/journal.pone.0021101

Tsai WT (2005) Service-oriented system engineering: a new paradigm. Proceedings of the 2005 I.E. International Workshop on Service-Oriented System Engineering, pp 3–6

Wong-Bushby I, Egan R, Isaacson C (2006) A case study in SOA and re-architecture at company ABC. Proceedings of the 39th Annual Hawaii International Conference on System Sciences

Yarmey L, Khalsa SL (2014) Building on the international polar year: discovering interdisciplinary data through federated search. Data Sci J 13(0):PDA79–PDA82

Zikopoulos P, Eaton C, deRoos D et al (2011) Understanding big data: analytics for enterprise class hadoop and streaming data. McGraw-Hill, New York