DATA CONSERVANCY PROVENANCE, CONTEXT, AND LINEAGE SERVICES: KEY COMPONENTS FOR DATA PRESERVATION AND CURATION

Matthew S. Mayernik[1*], Tim DiLauro[2], Ruth Duerr[3], Elliot Metsger[2], Anne E.Thessen[4], G. Sayeed Choudhury[2]

*[*1] National Center for Atmospheric Research, University Corporation for Atmospheric Research, Boulder, CO*
Email: mayernik@ucar.edu
[2] Johns Hopkins University Sheridan Libraries, Baltimore, MD
Email: {timmo, emetsger, sayeed}@jhu.edu
[3] National Snow and Ice Data Center, Boulder, CO
Email: rduerr@nsidc.org
[4] School of Life Sciences, Arizona State University, Phoenix, AZ
Email: annethessen@gmail.com

ABSTRACT

Among the key services that institutional data management infrastructures must provide are provenance and lineage tracking and the ability to associate data with contextual information needed for understanding and use. These functionalities are critical for addressing a number of key issues faced by data collectors and users, including trust in data, results traceability, data transparency, and data citation support. In this paper, we describe the support for these services within the Data Conservancy Service (DCS) software. The DCS provenance, context, and lineage services cross the four layers in the DCS data curation stack model: storage, archiving, preservation, and curation.

Keywords: Provenance, Lineage, Context, Research data, Data curation, Preservation, Infrastructure

1 INTRODUCTION

Digital data collections offer opportunities for new and integrative research. In fact, many types of research, including synthesis studies, longitudinal analyses, and global-scale investigations, require that data from multiple sources be discovered, accessed, and brought together for analysis. However, efficient discovery and open access to digital research data is impeded by a number of challenges given the way the science community currently manages their data (Overpeck, et al., 2011; Wolkovich, Regetz, & O'Connor, 2012). Among the most significant challenges is the fact that most researchers do not archive their data sets in organizational or disciplinary data repositories (Kuipers & van der Hoeven, 2009; Science Staff, 2011). Many researchers manage their data themselves, storing data on personal hard drives, lab servers, or portable storage devices. While there might be many reasons for this (Costello, 2009; Enke, et al., 2012), including convenience, flexibility, and control, most of the data managed by individual investigators will be lost over time due to physical degradation of the media on which they are stored, knowledge loss over time as investigators forget details about the data collection and analysis processes, and personal factors such as the eventual retirement or death of individual investigators (Michener, et al., 1997).

In order to mitigate this long-term data management burden on individual investigators, organizational and institutional support is required. A key problem, however, is that there is a lack of reliable data management infrastructure that can be deployed at an institutional level. Many research organizations, particularly universities, currently do not have data management infrastructures and services in place, and those infrastructures and services that do exist are often specific to a particular project or type of data and do not cover the entire range of services needed to curate data for the long term. Given the increasing emphasis on data sharing and enduring access currently being placed on data at both the national and international levels, this situation is becoming increasingly untenable. United States federal agencies are calling for data that led to a given research result to be more discoverable and usable for secondary purposes (Holdren, 2013; NASA, 2011; NOAA Environmental Data Management Committee, 2011), in order to ensure the transparency, traceability, and reusability of taxpayer-funded research. In addition, a

variety of international organizations such as the newly formed Research Data Alliance (https://rd-alliance.org/) are working to make research data more broadly available and interoperable.

One of the key services that institutional data infrastructures must provide to ensure data transparency, traceability, and reusability is provenance and context tracking (Waters & Garrett, 1996; CCSDS, 2012). Provenance and context tracking refers to the creation of a record that contains details of the background of an object, possibly including information on how an object was created, the processing steps it has gone through, when those creation or processing events took place, and what has happened to that object since its creation. Provenance functionalities are critical to addressing a number of key issues faced by data infrastructures: trust in data, results traceability, transparency of data modification or analysis processes, and data citation support.

In this paper, we describe two provenance-related services, dubbed the "provenance stream" and the "lineage service," within the Data Conservancy Instance, a data curation infrastructure designed to be a solution to institutional data challenges. In addition, we describe an initial set of capabilities for a Data Conservancy Instance to ingest and record provenance and contextual information from the external environment.

2 DATA PROVENANCE AND LINEAGE LITERATURE

Collecting provenance information about data resources within a data curation infrastructure encompasses many tasks and system components. A number of studies have enumerated the challenges of collecting provenance information and the necessary tasks involved (Gil, et al., 2010; Miles, et al., 2007; Moreau, et al., 2011).

2.1 A note on terminology

The terms "provenance" and "lineage" are often used interchangeably. Bose and Frew (2005) distinguished the two terms as follows: provenance refers to "the sources of query- and service based data processing results, while lineage connotes the processing history of a data product" (pg. 4). Missier, et al., (2008) discuss data "lineage" via examinations of "the graph of data dependencies that account for an output value produced during the course of a dataflow execution" (pg. 18). Most discussions of provenance, however, use the two terms interchangeably (see Simmhan, Plale, & Gammon, 2005, for an example). In this paper, we use "provenance" as the over-arching term while using "lineage" for the specific purpose to indicate graphs of dependencies between data resources. This is discussed more in Section 4.

2.2 Provenance motivations, benefits, and processes

The benefits of collecting provenance and lineage information about data resources are numerous, including communicating the quality of data and the suitability of data for particular uses and preventing misinterpretation by expert or non-expert users by communicating the processing steps that led to the creation of data products (Bose & Frew, 2005). The W3C created a Provenance Incubator Group with the task of outlining a roadmap for developing recommendations and standards related to the provenance of digital objects. The final report from this Incubator Group defined the provenance of a resource as "a record that describes entities and processes involved in producing and delivering or otherwise influencing that resource" (Gil, et al., 2010).

The idea that the provenance of digital objects should be manifested as a "record" is important. Within any given data infrastructure, provenance records must be generated in a standardized form across what might be heterogeneous data collections. Provenance information must also be made available and accessible to system functionalities in an efficient way as provenance records are pointless if they are not used to inform the system or the data user of the background and lineage of data resources. Thus, the motivations for recording provenance information must be clear before implementing specific provenance capture services (Groth, et al., 2012). Also, it should be noted that provenance records could be prospective or retrospective (Freire, et al., 2008). Prospective records document a process that must be followed to generate a given class of products whereas retrospective records document a process that has already been executed.

A number of processes are involved in collecting provenance information. Groth and Moreau (2009) outlined a number of characteristics of provenance records that enable the documentation of computational processes.

- Immutable - Provenance records should not change once created.
- Attributable - Responsibility for creating provenance documentation must be clear and should be transparent to others who are accessing and using data.
- Autonomously creatable - System components should autonomously create provenance information as processes are executed.
- Finalizable - Provenance information should be markable as the final representation of a completed process.
- Process reflecting - When compiled, provenance records should reflect how a distributed system executed overall.

Many artifacts might be relevant to tracking the provenance and context of particular data sets: data sources (instruments and platforms), calibration data, algorithms, people, computer hardware specifications, published papers, and abstract events (Tilmes, Yesha, & Halem, 2010). As noted by Moreau, et al., (2008) "[a] provenance query must be able to identify a data item with respect to a given documented event" (pg. 56). Provenance-related events will vary across systems but might include data processes, transformations, or migrations that make changes to particular data files or collections. Such system events have significant provenance implications and are thus important to document as part of provenance services.

The rest of this paper discusses how provenance, context, and lineage services have been implemented in the Data Conservancy data curation infrastructure.

3 ABOUT THE DATA CONSERVANCY

The Data Conservancy is a community organized around a shared technical infrastructure and a set of organizational services for data curation (Mayernik, et al., 2012). Data Conservancy Instances leverage the Data Conservancy Service (DCS) software stack, which provides a discipline-agnostic data curation infrastructure that can be deployed at organizational levels. An alpha version of the Data Conservancy software has been released as an open source package and can be downloaded at http://dataconservancy.org/software/downloads/. DC Instances have been deployed within the Johns Hopkins University (JHU) Data Management Services (http://dmp.data.jhu.edu/) to archive data that are produced via research grants received by JHU investigators and in the National Snow and Ice Data Center (NSIDC) to manage resources related to the Exchange for Local Observations and Knowledge of the Arctic (ELOKA) project. The ELOKA project is discussed more below.

The Data Conservancy developed a stack model that conceptualizes the key concepts of storage, archiving, preservation, and curation in relation to the DCS. This model has been useful for communicating with researchers who often use such terms interchangeably. The model is not intended to be a definitive statement but rather a reflection of our lessons learned through the R&D, prototyping, and implementation of the Data Conservancy and the associated operational or service environment within the Johns Hopkins University Data Management Services. The model is hierarchical in the sense that storage is necessary but not sufficient for archiving; archiving is necessary but not sufficient for preservation, and so on. Each layer above depends on the layer below.

At the lowest level of service in the data management stack model is storage that describes bits on disk, tape, or in the cloud with backup and restore services. Archiving (sometimes called bit-level preservation by others) focuses on data protection through actions or concepts such as replication, fixity, and/or identifiers. Preservation, for Data Conservancy, involves providing enough representation, context, metadata, fixity, and provenance information such that someone -- or some machine -- other than the original data producer can use and interpret the data. Curation refers to adding value to foster discovery and reuse, in particular re-use by communities other than the community that produced the data.

Table 1. Data Conservancy Service stack model

Layers	Characteristics	Implication for investigators	Implications relative to funding agency requirements
Curation	Adding value throughout life-cycle	Investigator benefits from advanced features •Feature Extraction •New query capabilities •Cross-disciplinary discovery	•Competitive advantage •New opportunities
Preservation	Ensuring that data can be fully used and interpreted	•Investigator has ability to use own data in the future (e.g., 5 yrs.) and share with other interested users	•Satisfies funder data management requirements •Competitive advantage
Archiving	Data protection including fixity, identifiers	•Provides identifiers for sharing, references, etc.	•Could satisfy most funding agency requirements
Storage	Bits on disk, tape, cloud, etc. Backup and restore	•Investigator responsible for: •Restoring •Sharing •Staffing	•Could be enough for now but not near-term future

The development of the DCS software stack used the Open Archival Information System (OAIS) reference model as a guide (CCSDS, 2012). Instances of the DCS offer an extensive set of capabilities, including: a preservation-ready infrastructure, customizable user interfaces, application programming interfaces (APIs) for many services, and a scalable storage back-end. The DCS also contains a feature extraction framework that allows data from multiple projects to be brought together based on particular data characteristics. Through this framework, feature extraction plug-ins can be written for any user-defined purpose. Some of these capabilities are common to data curation systems, such as ingest, storage, and search/browse, and some are unique to the Data Conservancy system, such as the feature extraction framework.

The Data Conservancy Instance employs multiple data models, differentiated by their role in the Instance. These models are shown in Figure 1. The *archival* data model used by the DCS is based on the PLANETS conceptual data model (http://www.planets-project.eu/). The PLANETS data model is compatible with the OAIS framework and with provenance metadata models like PREMIS (http://www.loc.gov/standards/premis/). The archival data model provides abstractions over the data in the Instance's custody, allowing various services within the DCS to interact with the data using shared semantics. The *business* data model, on the other hand, provides abstractions for services external to the DCS to interact with the data. A mapping layer translates instances of a business model (i.e., business objects) to an equivalent archival instance (colloquially, a business object's "archival representation") and back. The mapping layer de-couples the data models, allowing them to evolve independently as business and archival needs dictate. Typically a single business object is represented as a graph of archival objects, headed by (i.e., the root) a *deliverable unit* (a specific type of entity in the archival data model). Stable identifiers are used to maintain a reference between the business object and the root *deliverable unit*; even as business objects are modified over time, the reference to the business object's archival representation never changes.

Instances of the archival model are immutable; once archived, the properties of archival objects cannot be modified. Therefore, modifications to business objects are expressed using a special relationship between archival object graphs, where one graph may represent the state of a business object at time t, and its successor graph represents the modified state of the business object at time $t+n$. This relationship provides the basis for the lineage of archival object graphs.

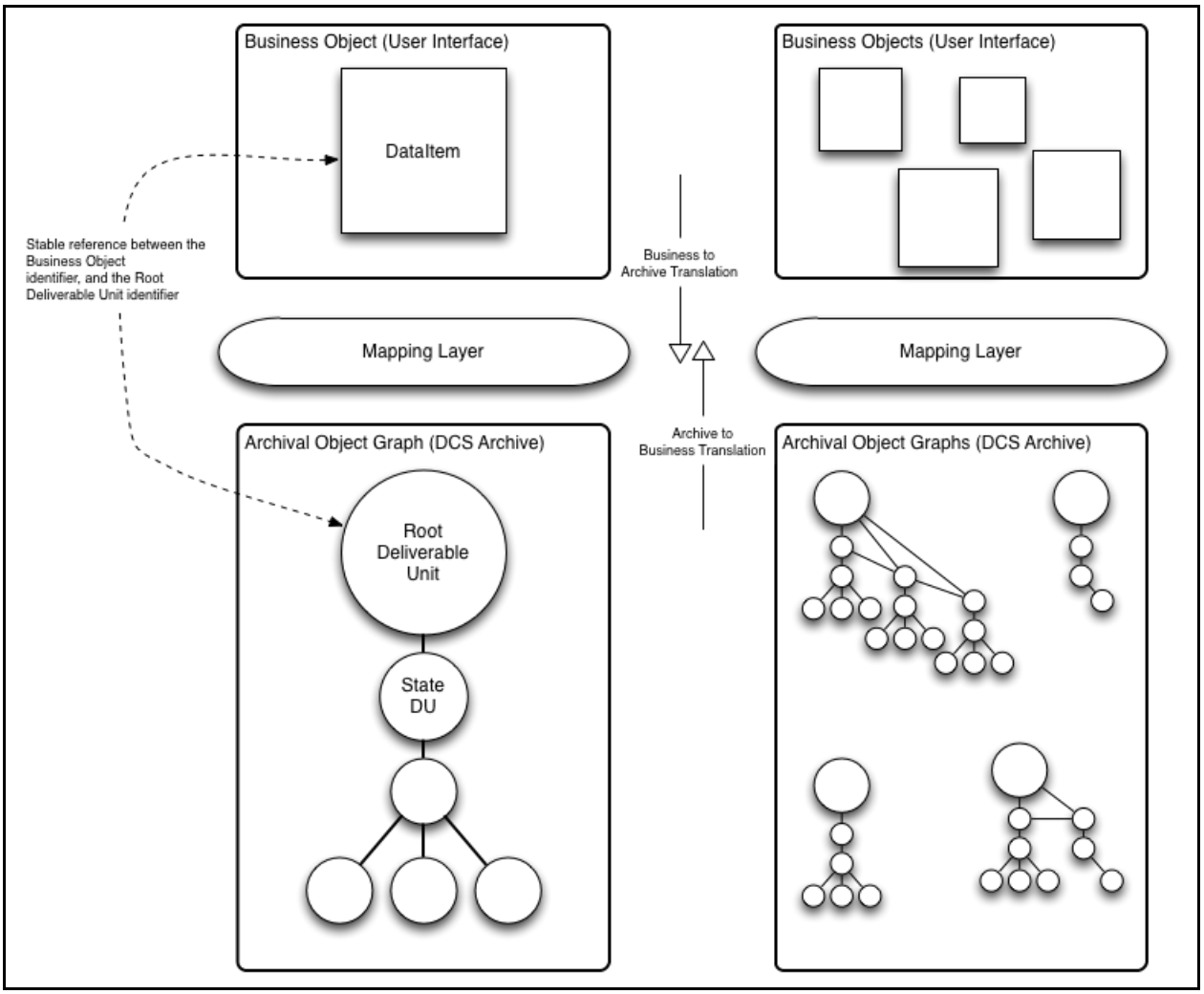

Figure 1. Data Conservancy data models in practice

As shown in Figure 2 below, the DCS software architecture contains multiple layers. Each layer communicates with the layers immediately above and below it. Applications, such as the Reference User interface depicted here or batch load processes, can be built on top of the DCS core services or the business layer, layers which mirror the data model implementations. The DCS core itself is a layered component as described below:

- API layer – The API layer provides the specifications for how services are accessed and invoked. The APIs are invoked via HTTP requests.
- Services layer – The services layer consists of services that are invoked as needed by the applications via the APIs. These services include ingest, indexing, and search and access.
- Archiving layer – The archival storage API is the interface to the archival services and is used to put data into the archive and to bring data from the archive to the users.

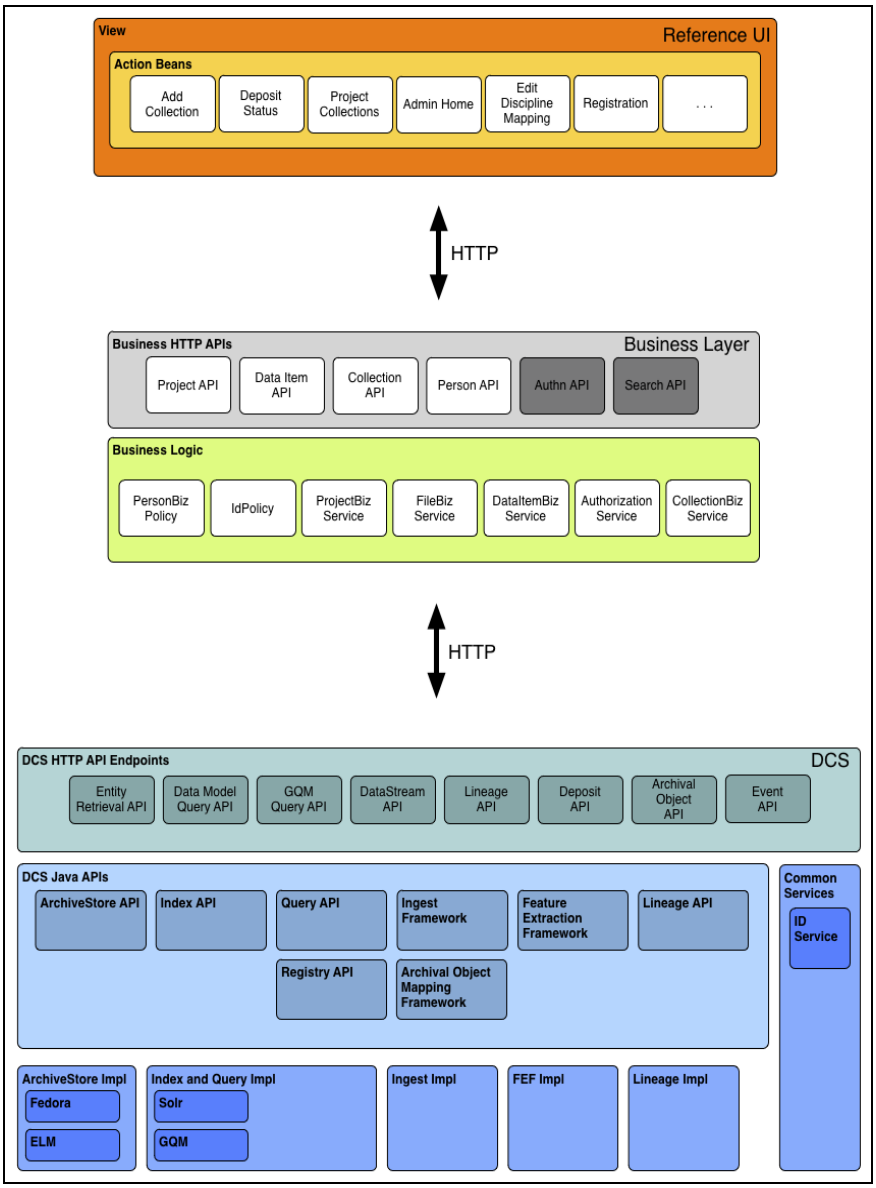

Figure 2. Data Conservancy Service (DCS) software architecture

Within this layered stack, a number of components provide provenance and context functionalities, including multiple services in the service layer as well as the lineage and ingest APIs.

4 PROVENANCE, CONTEXT, AND LINEAGE IN THE DC SOFTWARE STACK

The development of the DCS has been driven by a combination of user needs and infrastructure requirements. The provenance, context, and lineage-related capabilities and services reflect both of those development inputs. This section discusses the DC provenance stream and the DC lineage services, followed by a section describing the initial suite of capabilities for associating objects with provenance and contextual information that pre-dates their ingest into the DCS.

We use the distinct names “provenance stream" and "lineage services" because they are distinct functionalities from a technical standpoint. These names are descriptive in use, allowing us to distinguish the two capabilities during the development process. At heart, however, they both serve the larger goal of collecting provenance information in order to represent the broader context that surrounds data ingested into the DCS.

Multiple infrastructure and user-centric motivations led to the development of the DC provenance stream and lineage services. The need to capture provenance-related system events, such as format migrations, was recognized from the beginning stages of the DCS development (Tilmes, Yesha, & Halem, 2010). The DCS lineage service, on the other hand, was motivated by user needs, namely the need to be able to deposit a new version of a resource into the repository (see also Gray, et al., 2002). In both cases, the information collected is retrospective provenance, namely the computational steps that have been executed by the system (Freire, et al., 2008). More recently, capabilities to ingest information about the provenance and context of an object prior to its ingest into the DCS have been added.

4.1 Provenance stream

In the DCS, the provenance functionality is better characterized as a provenance stream than a provenance service per se. The provenance stream is initiated as part of the multi-step data ingest process. Provenance information is recorded whenever system *events* occur. *Events* in the ingest pipeline are particular checks and processes that are run over every ingested data submission. For example, one such *event* is a characterization event, which determines the mime type of the ingested files. Another ingest *event* is an internal identifier assignment. Each *event* is a representation of a completed process, such as a format characterization, fixity check, or identifier assignment.

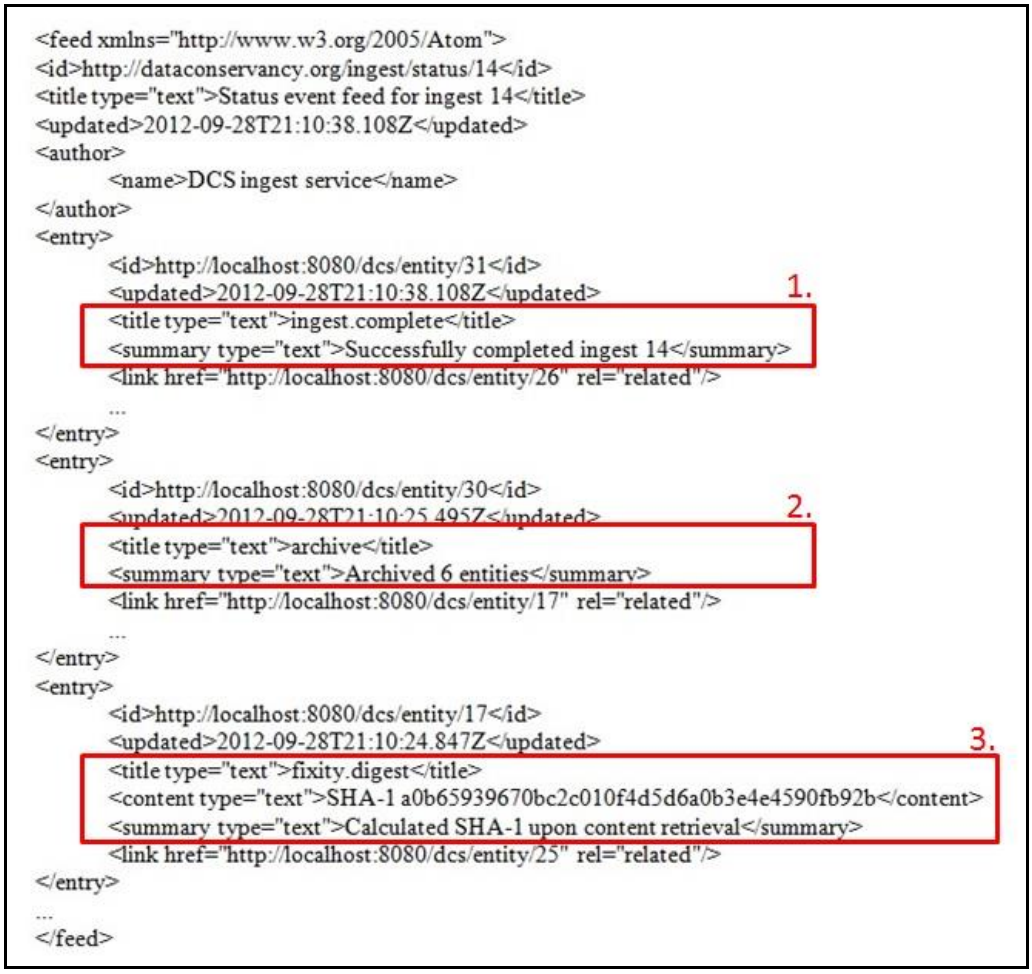

Figure 3. Snippet of DCS provenance stream in XML, showing: 1) successful ingest acknowledgement, 2) number of entities archived, and 3) record of Secure Hash Algorithm (SHA) calculation.

The provenance stream records all of the events that occur on an item as it is ingested into the archive. This record of events depicts contextual factors that help to ensure proper ingest and any issues that occurred during the process.

The system generates an XML document that serializes these provenance events for each object archived by the system. This XML document is called that object's provenance stream and is available via a permanent URL. An XML provenance stream is initiated with the *event* that a deposit request itself has been received. The ingest process records *events* at each step until ingest has completed. The ingest process has three different possible status states: pending (upon the receipt of a submission), in progress, and completed. At the end, the ingest success or fail is declared in the provenance stream. There is a distinct event generated that indicates the final status. At any time, the provenance XML stream can be requested as an Atom feed by the user or system administrator.

The example in Figure 3 shows portions of a provenance feed in Atom XML. (Note: this is a demonstration example; the "localhost" identifiers will not work if queried.) Figure 3 shows three examples of events that are recorded in the DCS provenance stream during the process of ingesting items into the archive. Looking at these in reverse order, event #3 records the event of applying the Secure Hash Algorithm (SHA-1) to produce a fixity calculation that is used to ensure the integrity of ingested files over time. Event #2 records the successful archiving of six entities as part of this ingest process, and event #1 shows an acknowledgement that the ingest process has completed successfully. If an error had occurred during the ingest process that caused the ingest to fail, the provenance stream would have recorded the error as well as the fact that the ingest did not complete successfully.

Within the DCS, individual entities, including *deliverable units* and *events*, are immutable once deposited. Because entities are immutable, changes to resources take place through the deposition of new iterations. The ability to deposit multiple iterations of a particular resource is provided by the lineage service.

4.2 Lineage service

The motivating questions for the lineage service are around how to represent changes to data resources and relations between data resources. To address these questions, the lineage service provides semantics to identify and represent new versions of deposited resources. At heart, the lineage service is a mechanism for recording relationships between two *deliverable units*. Formally, the data model allows the user to declare any relationship, but the relationship type recognized and leveraged by the DC system currently is "isSuccessorOf". All relations between versions are sequential. When adding a new version of a *deliverable unit*, Version 2 will have a relationship "isSuccessorOf" pointing to Version 1, as shown in Figure 4.

Recall that the root *deliverable unit* is the head of an archival object graph representing a business object. The state (i.e., set of properties) of the business object is represented by a subset of the archival graph, headed by a state *deliverable unit*. Therefore, a lineage of a business object state is expressed by "isSuccessorOf" relationships between state *deliverable units*. The "parent" relationship shown in Figure 4 maintains the connection between the root *deliverable unit* and the successor *deliverable units* in each lineage graph. The *deliverable units* that participate in a given lineage share a unique, immutable lineage identifier. In Figure 4, there is one participant in the lineage for version 1, two participants for version 2, and three participants for version 3. The lineage identifier is the same for all versions; only the number of participants in the lineage changes. Requesting the lineage identifier encoded as a HTTP URL returns a representation of all of the *deliverable units* in the identified lineage. The lineage API also supports methods for obtaining the head or tail of a lineage or querying the lineage after a particular date or within a time range.

Conceptually, the business layer in the DCS is responsible for determining the semantics of "version". Consider an Item business object that carries a "description" property: a short, human-readable, textual description of the object. When an administrator updates the "description" property of the *item* (for example, to correct a typographical error), the business layer must decide if the updated Item represents an entirely new *item* or if it represents a new version of an existing *item*. If the business layer determines the update represents a new version, it will construct an archival representation of the *item* that references its successor in the lineage.

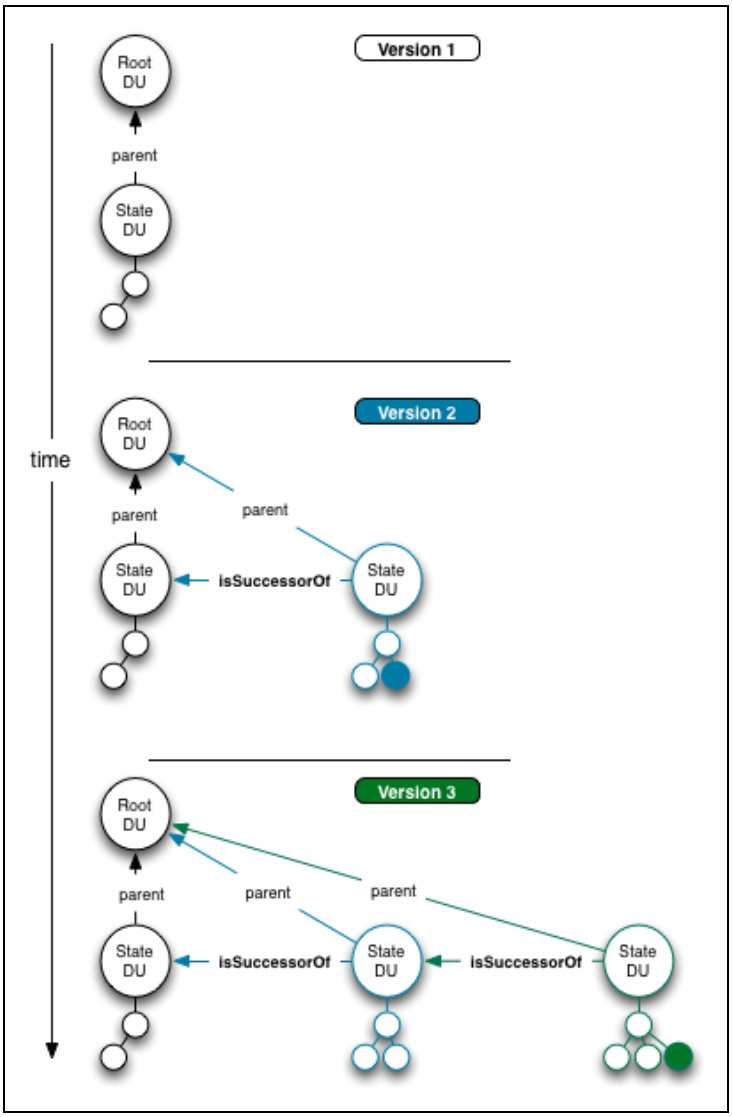

Figure 4. Lineage established over time, expressed by relationships between archival entities

4.3 Support for ingest of provenance, context, and lineage information

While the provenance and lineage services described above can be used to capture system events that occur to an object starting from the point of ingest into the DCS as well as allowing multiple versions of an object to be archived, the remaining piece of the provenance, context, and lineage puzzle that has yet to be described are the DCS capabilities to ingest and record information about the provenance and context of an object prior to its ingest into the DCS. In support of these needs the DCS can either ingest the needed contextual information as information objects in their own right and associate them with existing objects or collections of objects within the DCS, or users are allowed to define relationships between objects held within the DCS itself and those held elsewhere, for example by another repository. In either case, the key ability is the ability to define relationships between items within the DCS and the external world.

As mentioned previously, the only relationship supported in the alpha release of the DCS was the "isSuccessorOf" relation used in the lineage service. In the beta release of the DCS a number of additional relationship types will be supported that can be used to more completely reflect the provenance, context, and lineage of an object held within the DCS. These relationships are described in Table 2 below. It should be noted that most DCS relationships are symmetric. That is, for each relationship of the form Target "isRelatedTo" Source, there is a reverse relationship of

the form Source "hasRelationWith" Target. In Table 2 below, the DCS Relationship column defines the relationships that end users might see when choosing a type of relationship to associate between two objects; while the Underlying Relationships column contains a set of additional relationships from other standard ontologies and vocabularies that are used under the hood to assure that relationships defined within the DCS system can be re-used outside the DCS context. As Table 2 shows, the DCS makes use of the W3C PROV approach for modeling provenance information (Groth & Moreau, 2013). The PROV relationships that are supported have been renamed in the DCS because end users may not understand the PROV terminology. In renaming the relationships, the complexity of the PROV model is hidden under the DCS hood in order to present users with clear concepts and labels. One last note on the W3C PROV model is that members of the Data Conservancy community are the current chairs of an interest group under the auspices of the Research Data Alliance (RDA) focused on provenance data models (see https://www.rd-alliance.org/internal-groups/research-data-provenance.html). Multiple provenance data models and standards exist and are in use at both general and disciplinary levels, including W3C PROV (Ding, et al., 2010). This RDA interest group will examine the merits of the various models and standards and consider the interoperability of their outputs. So while the DCS does not embrace the full scope of the W3C PROV model at this time, we are fully engaged in the broader community effort to examine the available and relevant options for representing this information.

As the other Underlying Relationships depicted in Table 2 show, the DCS provenance ontology is extensible to a wide variety of data-related relevant ontologies, ensuring that the DCS approach can be understood and used by other groups that also use those ontologies. In addition, as noted above, users can declare any relationship between entities that they like beyond those in this table. The current DCS, however, within its own reasoning is only capable of acting upon or expressly using the relationships in Table 2. Additional relationships will be incorporated into subsequent versions of the DCS based on user and Data Conservancy Instance owner requests.

Table 2. Additional relationships supported in the beta release of the DCS software system

DCS Relationship	Underlying Relationships	Source can be	Target can be
isMetadataFor	dc:description dc:conformsTo	File, Data Item, external URI	Data Item, Collection, Project
isAncillaryDataFor	prov:used	File, Collection, Data Item, external URI	Collection, Data Item
isProjectFor	prov:hadMember dc:relation	Project	Collection, Data Item
isUsageAgreementFor	dc:licenseDocument	Data Item	Project, Collection, Data Item
isAccessRestrictionsFor	dc:accessRights	Data Item	Project, Collection, Data Item
isDocumentationAbout	dc:description	File, Data Item, external URI	Data Item, Collection, Project
isDerivedFrom	prov:wasDerivedFrom prov:Derivation dc:source data:DataProduct	File, Data Item, Collection	Data Item, Collection, external URI
isMemberOf	prov:hadMember dc:isPartOf	File, Data Item, Collection	Data Item, Collection, Project

4.4 Provenance, context, and lineage example

The Exchange for Local Observations and Knowledge of the Arctic (ELOKA) project provides an example of how the provenance, context, and lineage service the DCS provides facilitates the curation of diverse sets of resources. ELOKA provides data management and user support for observations and knowledge of the Arctic that come from local communities and scientific studies. The National Snow and Ice Data Center, in Boulder, CO, is deploying a Data Conservancy Instance (DC-I) to manage and curate a diverse set of ELOKA resources. Among these resources are analog items that have been digitized. In particular cases, the digitized items have then been overlaid with other digital information to create merged products – typically composite maps. Curating these products requires managing the original and the digitized versions of each product as well as the merged products that result. ELOKA

archives the original materials in the Roger G. Barry Archives and Resource Center and the digital versions in the DC-I. As digitized versions of documents are uploaded into the DC-I, the DCS relationship capabilities record the derivation record of the data as depicted in Figure 5 below while the lineage services are used to record new versions of any given product. Along with the provenance service, which provides a record of everything that happens to a particular object once it is ingested, a complete provenance, context, and lineage record can be maintained.

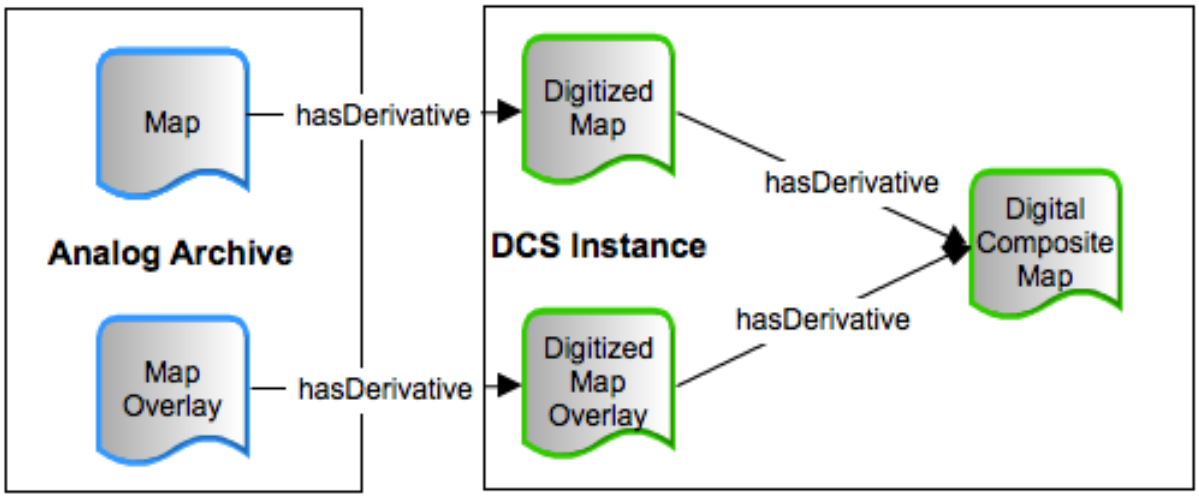

Figure 5. An example set of objects related by a derivation chain.

5 CONCLUSION

In order to ensure that data remain understandable and usable, data infrastructure must manage relationships and processes in addition to the resources themselves. Provenance, context, and lineage services provide scalable ways to trace and manage relationships between numerous resources and to track the processes used to manipulate, transform, and preserve resources. Within the Data Conservancy Service, the provenance functionality documents internal processes, the lineage service establishes linkages between successive versions of submitted resources, and the ability to record relationships between resources allow users to associate provenance and contextual information with a resource. These services reflect the characteristics of provenance ready computational systems noted by Groth, et al. (2012). The results of completed processes are compiled into the provenance and lineage streams as system events by the DCS, reflecting how the processes were executed and their success or failure. DC provenance and lineage streams are autonomously created by the DCS as system events occur and are immutable once created.

From a service stack perspective, the DCS provenance and lineage services provide functionalities that cross the four layers in the Data Conservancy Service stack model: storage, archiving, preservation, and curation.

- Storage - The DCS begins storing provenance information upon ingest along with the resources themselves. The stream of DCS-generated provenance information can then be leveraged to assess the success and failure of individual processes and of sets of processes, such as the ingest process.
- Archiving - The DCS creates identifiers for individual entities as they are ingested and creates a lineage identifier that identifies a set of related resources. The lineage identifier also provides the mechanism by which lineage information can be queried and retrieved.
- Preservation - Capturing system events as provenance and lineage information is one component of a preservation-ready system, in that these streams allow people and machines other than the original data producer to use and interpret the data. Providing users the capability of associating a resource in defined ways with other information irrespective of whether that information is held within the DCS or not maximizes the chances that all of the information needed by future users will be available.

- Curation - The DCS allows additional features to be built on the lineage service and the ability to define relationships between resources both within and outside of the DCS. The lineage service will underpin user interfaces that allow users to navigate through changes in versions and understand an object's history and context. This flexibility and customizability is a design feature of the software. In the alpha release of the Data Conservancy software, the reference user interface hides the details of the lineage service from the user.

Provenance, context, and lineage functionalities are essential components of infrastructure for digital research data. The Data Conservancy stack model helps to identify where provenance and lineage functionalities fit within the storage, archiving, preservation, and curation service layers.

As noted in Section 3 above, the initial release of the DCS is available as an open source package on the Data Conservancy web site. A roadmap for future work and future releases of the DCS is in progress, and funding models are being explored to support this future work. The funding model will likely include a combination of resources including Johns Hopkins University, additional grants, community contributions, and potential fee-based services. Future extensions of the work described in this paper will investigate the consequences of allowing users to assert pre-existing provenance events about objects during accession by the DCS archive, in order to, for example, represent format conversions that took place prior to objects being uploaded to the DCS. Other extensions will explore the potential for a unified *event* model between the business layer and the archive and develop an understanding of how preservation actions may affect the lineage of archival object graphs.

6 ACKNOWLEDGEMENTS

The Data Conservancy is funded by the National Science Foundation under grant number OCI-0830976. Funding for the Data Conservancy and the Johns Hopkins University Data Management Services is provided by the JHU Sheridan Libraries. We acknowledge contributions from our Data Conservancy colleagues.

7 REFERENCES

Bose, R. & Frew, J. (2005) Lineage retrieval for scientific data processing: a survey. *ACM Computing Surveys 37*(1), 1–28. Retrieved from the World Wide Web November 13, 2013: http://dx.doi.org/10.1145/1057977.1057978

Consultative Committee for Space Data Systems (CCSDS). (2012) *Reference Model for an Open Archival Information System (OAIS), Recommended Practice*, Issue 2, CCSDS 650.0-M-2. Retrieved from the World Wide Web November 13, 2013: http://public.ccsds.org/publications/archive/650x0m2.pdf

Costello, M.J. (2009) Motivating online publication of data. *BioScience 59*(5), 418-427. Retrieved from the World Wide Web November 13, 2013: http://www.jstor.org/stable/10.1525/bio.2009.59.5.9

Ding, L., Bao, J., Michaelis, J.R., Zhao, J., & McGuinness, D.L. (2010) Reflections on provenance ontology encodings. In D.L. McGuinness, J.R. Michaelis, & L. Moreau (Eds.), *Provenance and Annotation of Data and Processes* (Lecture Notes in Computer Science, Vol. 6378, pp. 198–205), Springer Berlin Heidelberg. Retrieved from the World Wide Web November 13, 2013: http://dx.doi.org/10.1007/978-3-642-17819-1_22

Enke, N., Thessen, A.E., Bach, K., Bendix, J., Seeger, B., & Gemeinholzer, B. (2012) The user's view on biodiversity data sharing. *Ecological Informatics 11*, 25-33. Retrieved from the World Wide Web November 13, 2013: http://dx.doi.org/10.1016/j.ecoinf.2012.03.004

Freire, J., Koop, D., Santos, E., & Silva, C.T. (2008) Provenance for computational tasks: a survey. *Computing in Science & Engineering 10*(3), 11–21. Retrieved from the World Wide Web November 13, 2013: http://dx.doi.org/10.1109/MCSE.2008.79

Gil, Y., et al. (2010) *Provenance XG Final Report: W3C Incubator Group Report*. World Wide Web Consortium (W3C). Retrieved from the World Wide Web November 13, 2013: http://www.w3.org/2005/Incubator/prov/XGR-prov-20101214/

Gray, J., Szalay, A.S., Thakar, A.R., Stoughton, C., & vandenBerg, J. (2002) Online scientific data curation, publication, and archiving. *Proc. SPIE 4846, Virtual Observatories*, 103. Retrieved from the World Wide Web November 13, 2013: http://dx.doi.org/10.1117/12.461524

Groth, P., Gil, Y., Cheney, J., & Miles, S. (2012) Requirements for provenance on the web, *International Journal of Digital Curation 7*(1), 39-56. Retrieved from the World Wide Web November 13, 2013: http://dx.doi.org/10.2218/ijdc.v7i1.213

Groth, P. & Moreau, L. (2009) Recording process documentation for provenance. *IEEE Transactions on Parallel and Distributed Systems 20*(9), 1246–1259. Retrieved from the World Wide Web November 13, 2013: http://dx.doi.org/10.1109/TPDS.2008.215

Groth, P. & Moreau, L. (Eds.) (2013) *PROV-Overview: An Overview of the PROV Family of Documents.* W3C Working Group Note 30 April 2013. Retrieved from the World Wide Web November 13, 2013: http://www.w3.org/TR/2013/NOTE-prov-overview-20130430/

Holdren, J.P. (2013) *Memorandum for the Heads of Executive Departments and Agencies: Increasing Access to the Results of Federally Funded Scientific Research*. Wash., D.C.: Executive Office of the President, Office of Science and Technology Policy, Feb. 22, 2013. Retrieved from the World Wide Web November 13, 2013: http://www.whitehouse.gov/sites/default/files/microsites/ostp/ostp_public_access_memo_2013.pdf

Kuipers, T. & van der Hoeven, J. (2009) *Insight into digital preservation of research output in Europe: Survey report*, PARSE.insight. Retrieved from the World Wide Web November 13, 2013: http://www.parse-insight.eu/downloads/PARSE-Insight_D3-4_SurveyReport_final_hq.pdf

Mayernik, M.S., Choudhury, G.S., DiLauro, T., Metsger, E., Pralle, B., Rippin, M., & Duerr, R. (2012) The Data Conservancy Instance: infrastructure and organizational services for research data curation. *D-Lib Magazine 18*(9/10). Retrieved from the World Wide Web November 13, 2013: http://dx.doi.org/10.1045/september2012-mayernik

Michener, W.K., Brunt, J.W., Helly, J.J., Kirchner, T.B., & Stafford, S.G. (1997) Nongeospatial metadata for the ecological sciences. *Ecological Applications 7*(1), 330-342.

Miles, S., Groth, P., Branco, M., & Moreau, L. (2007) The Requirements of using provenance in e-science experiments. *Journal of Grid Computing 5*(1), 1–15. Retrieved from the World Wide Web November 13, 2013: http://dx.doi.org/10.1007/s10723-006-9055-3

Missier, P., Belhajjame, K., Zhao, J., Roos, M., & Goble, C. (2008) Data lineage model for Taverna workflows with lightweight annotation requirements. In *Provenance and Annotation of Data and Processes* (Lecture Notes in Computer Science, Vol. 5272, pp. 17–30). Springer Berlin Heidelberg. Retrieved from the World Wide Web November 13, 2013: http://dx.doi.org/10.1007/978-3-540-89965-5_4

Moreau, L., Clifford, B., et al. (2011) The Open Provenance Model core specification (v1.1). *Future Generation Computer Systems 27*(6), 743–756. Retrieved from the World Wide Web November 13, 2013: http://dx.doi.org/10.1016/j.future.2010.07.005

Moreau, L., Groth, P., et al. (2008) The provenance of electronic data. *Communications of the ACM 51*(4), 52–58. Retrieved from the World Wide Web November 13, 2013: http://dx.doi.org/10.1145/1330311.1330323

National Aeronautics and Space Administration (NASA). (2011) *Data & Information Policy.* Retrieved from the World Wide Web November 13, 2013: http://science.nasa.gov/earth-science/earth-science-data/data-information-policy

NOAA Environmental Data Management Committee. (2011) *NOAA Data Sharing Policy for Grants and Cooperative Agreements: Procedural Directive, Version 1.0.* Retrieved from the World Wide Web November 13, 2013: https://www.nosc.noaa.gov/EDMC/DAARWG/docs/EDMC_PD-Data_Sharing_Policy_v1.pdf

Overpeck, J.T., Meehl, G.A., Bony, S., & Easterling, D.R. (2011) Climate data challenges in the 21st century. *Science 331*(6018), 700–702. Retrieved from the World Wide Web November 13, 2013: http://dx.doi.org/10.1126/science.1197869

Science Staff (2011) Challenges and opportunities. *Science 331*(6018), 692–693. Retrieved from the World Wide Web November 13, 2013: http://dx.doi.org/10.1126/science.331.6018.692

Simmhan, Y.L., Plale, B., & Gannon, D. (2005) A survey of data provenance in e-science. *ACM SIGMOD Record 34*(3), 31. Retrieved from the World Wide Web November 13, 2013: http://dx.doi.org/10.1145/1084805.1084812

Tilmes, C., Yesha, Y., & Halem, M. (2010) Tracking provenance of earth science data. *Earth Science Informatics 3*(1-2), 59–65. Retrieved from the World Wide Web November 13, 2013: http://dx.doi.org/10.1007/s12145-010-0046-3

Waters, D. & Garrett, J. (1996) *Preserving Digital Information.* Report of the Task Force on Archiving of Digital Information. Washington, DC: CLIR. Retrieved from the World Wide Web November 13, 2013: http://www.clir.org/pubs/reports/pub63watersgarrett.pdf

Wolkovich, E.M., Regetz, J., & O'Connor, M.I. (2012) Advances in global change research require open science by individual researchers. *Global Change Biology,18*(7), 2102–2110. Retrieved from the World Wide Web November 13, 2013: http://dx.doi.org/10.1111/j.1365-2486.2012.02693.x

(Article history: Received 18 July 2013, Accepted 9 November 2013, Available online 17 November 2013)