ELSEVIER

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 00 (2010) 1–9

Procedia Computer Science

International Conference on Computational Science, ICCS 2010

Distinguishing Provenance Equivalence of Earth Science Data

C. Tilmes[a,*], Ye. Yesha[b], M. Halem[b]

[a] *NASA Goddard Space Flight Center*
[b] *University of Maryland, Baltimore County*

Abstract

Reproducibility of scientific research relies on accurate and precise citation of data and the provenance of that data. Earth science data are often the result of applying complex data transformation and analysis workflows to vast quantities of data. Provenance information of data processing is used for a variety of purposes, including understanding the process and auditing as well as reproducibility. Certain provenance information is essential for producing scientifically equivalent data. Capturing and representing that provenance information and assigning identifiers suitable for precisely distinguishing data granules and datasets is needed for accurate comparisons. This paper discusses scientific equivalence and essential provenance for scientific reproducibility. We use the example of an operational earth science data processing system to illustrate the application of the technique of cascading digital signatures or "hash chains" to precisely identify sets of granules and as provenance equivalence identifiers to distinguish data made in an an equivalent manner.

Keywords: provenance, equivalence, reproducibility, data identifiers, data citations

1. Introduction

Provenance can be used for many purposes. For the Earth Science Data Processing domain, these include *understanding* of data and analyses, *auditing* , and *anomaly resolution*. Various facts can be semantically related to the artifacts, describing their provenance and other metadata about the data production and analysis. This paper will consider precise data identification and *reproducibility*.

1.1. Some Definitions

Some of these terms get used differently in different contexts, so let us define our use of them for this work.

Various people and organizations have attempted to nail down a concrete definition for *Provenance*. For reference, we'll copy a working definition from the W3C Provenance Incubator Group:

Definition 1.1. ***Provenance*** *refers to the sources of information, such as entities and processes, involved in producing or delivering an artifact.*[1]

*Corresponding author
Email address: Curt.Tilmes@nasa.gov (C. Tilmes)

[1] http://www.w3.org/2005/Incubator/prov/wiki/What_Is_Provenance

We include the basic workflow information, input files and processes, but also other ancillary aspects of the process, including the build environment, compilers and libraries. We also include the compute environment, such as operating system and computer hardware. [1].

Definition 1.2. ***Granularity*** *refers to the differentiation between individual portions of a whole dataset.* ***Granule*** *describes a file or set of related files corresponding to an individually identifiable portion of data and a* ***Granule Key*** *is used to distinguish multiple granules of the same type. [2]*

In the earth science domain, the identification of a granule is usually (but not always) related to the geospatial and temporal coordinates of the data contained in it.

Definition 1.3. *A* ***Granule Identifier*** *is a globally unique identifier for a specific granule.*

Every granule produced is assigned that identifier and it serves as the nexus for organizing information about that granule. When considering reproducibility, another entity can make the same granule in the same way, but since they are made at a different time by a different agent, each granule must be assigned a unique identifier so that provenance information can be attached to the right granule.

1.2. Equivalence and Reproducibility

Things can get confusing when considering reproducibility when two different and distinct instances of the 'same' granule are made in precisely the 'same' way such that they can reproduce the same science. We'll define a few terms to tease out the nuances of the equivalence we are discussing.

Definition 1.4. *For two granules of data to be* ***Perfectly Identical****, they must not only have identical contents, but also identical identifiers and identical provenance. If two otherwise identical granules do not share identical provenance, they are not perfectly identical.*

Given the definitions above, two granules will have the same *Granule Identifier* if and only if they are *Perfectly Identical*.

Definition 1.5. ***Scientifically Identical*** *refers to files (and more generally granules) where the data contents are bit-for-bit exactly the same.*

We use this term even if distinct granules were independently produced in a different time, place, or manner by different people. Based on our definition of provenance, such granules clearly have distinct provenance, but if their contents are equal, they are considered scientifically identical. Each such granule must have a distinct identifier that can refer to it so their particular provenance can be represented and referred to, but they can always be used interchangably in a scientific investigation. In practice, we find very few scientific processes and circumstances are capable of reliably producing such identical files. We use this term mainly to show the contrast with what we are actually capable of.

Definition 1.6. ***Scientifically Equivalent*** *refers to granules that are sufficiently similar that their use in a scientific investigation would result in the same results or conclusions.*

This is a somewhat more loose equivalence class than perfectly identical. For example, the processing could have been performed on slightly different hardware, or with a different level of compiler optimization. These changes could lead to very slight differences, while maintaining sufficient equivalence.

Of course, since it is impossible to foresee every possible future scientific investigation it is difficult or impossible to prove perfect scientific equivalence. It is more useful to limit the use of this term to a particular scientific domain, and show that duplicating certain aspects of the provenance of one granule will produce a scientifically equivalent granule for a given purpose.

Definition 1.7. ***Scientifically Reproducible*** *refers to a process which is capable of reproducing granules that are Scientifically Equivalent to the original granules and* ***Scientific Reproducibility*** *is the extent to which a process is Scientifically Reproducible.*

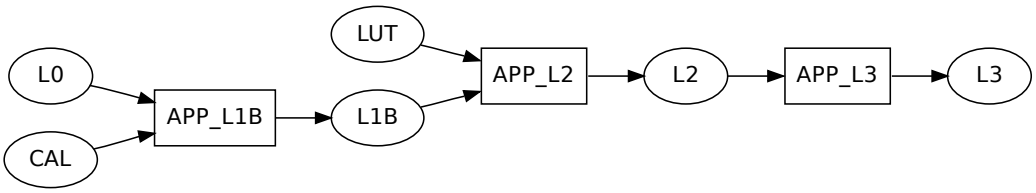

Figure 1: Simplified Ozone Processing Flow

Again, this is difficult to prove definitively. For example, if you were to process 1000 granules on computer *A*, then attempt to reprocess those same granules on computer *B*, comparing the results scientifically could lead you to believe that the process was reproducible, but it could fail due to some bug on granule 1001. Scientific Reproducibility is therefore a goal for a system or process, not a guarantee.

Some processes are chaotic in that very slight differences in processing are compounded producing drastically different results. We can apply sensitivity analyses to assess this characteristic and help determine if a process is suitably reproducible.

We also distinguish the related concepts of *Replication*—our own ability to repeat our processing and produce scientifically equivalent data from that of *Reproduction*—representing and conveying sufficient information about our process to an independent party so ***they*** can produce scientifically equivalent data.

This independent reproducibility provides needed credibility to results. Certain results from the earth science domain have been under fire for lack of transparency. Semantic provenance representation sufficient to enable independent reproducibility of scientifically equivalent data, and ultimately leading to the same scientific conclusions are critical for addressing that lack of transparency and increasing the credibility of our scientific results. [3]

Definition 1.8. ***Essential Provenance*** *are those portions of the provenance information of a granule that must be equivalent for a given process to reproduce that granule.*

This is used to distinguish elements of provenance that are useful for other purposes such as the understanding of a particular process, or auditing a data processing system from those specific elements essential for reproducibility.

For example, we capture the hostname of the computer where data processing was performed and the date/time the process occured. These don't really contribute to understanding of the process, but can increase credibility of results through auditing and assist in anomaly investigation. It isn't, however, essential to replicate these provenance elements for reproducibility.

Some provenance elements are clearly essential. For example, raw measurements captured by a satellite sensor that were fed into a process that produced a specific granule. It would be impossible to reproduce the resulting granule without using those precise inputs.

Other provenance elements may fall into a grey area. For example, a program may produce different answers depending on the hardware it is run on. For that particular process, the hardware (or some aspect of the hardware) could be essential. A goal of a science software developer should be to develop software where the correctness is less dependent on such provenance elements. This can increase the reproducibility of the process. Again, sensitivity analyses and portability testing can determine and help minimize the essential elements.

2. Use Case Scenario: Ozone Data Processing

This section will summarize a greatly simplified version of the data processing of ozone from satellite data [4] shown in figure 1. These data ultimately result in several figures related to the ozone hole and overall ozone coverage. They are provided annually to the Intergovernmental Panel on Climate Change (IPCC) and used to fulfill U.S. treaty obligations under the Montreal Protocol.

A sensor on a satellite captures measurements of backscattered sunlight from the earth. These raw ("level 0" or L0) data are transmitted to a receiving station and delivered to the processing system in 2 hour granules. The data are only captured on the sunlit side of each orbit of the spacecraft, so the easiest granularity for the scientists to work with is a single contiguous orbit per granule. The data undergoes a calibration and geolocation process ("level 1B" or L1B) that applies various calibration techniques and ultimately produces a level 1B granule for each orbit. The level 1B data go through various retrieval algorithms to determine specific geophysical parameters at level 2 (L2). In this case, the total amount of ozone in each vertical column of atmosphere. All of the orbits for each day are then summarized into a daily file (L3). Finally, considering the daily files for a year produces various averages, minima, and maxima for the annual report. These are used to monitor long term trends.

Provenance information is captured and archived throughout this process. This includes simple workflow information (as shown in the diagram), but also auditing information such as the host that performed each process, the time it started and stopped, etc. In particular the version of the algorithms, look up tables, and calibration files are critical, both for understanding and for reproducibility. [5]

Several aspects of this process make it an interesting real world case. One, it is an ongoing process—every day more data come in and get processed. Two, the scientists are still working on the process. This means they periodically analyze the data and find a problem and want to make a change. This could be a change to the calibration if they notice something happening with the instrument performance, or a change in the algorithms themselves. Another situation that occasionally arises are operational issues that lead either to a delay of data, or redelivery of level 0 data (e.g. the original data were corrupted or incomplete).

We have two (major) approaches to incorporating such changes. One, we can simply upgrade the system in place. This results in past granules remaining as is, and future granules being processed somewhat differently. The second approach is to start over from the beginning of the mission and reprocess all of the old data in the the new manner. This is typically used for more major changes, so the data don't exhibit discontinuities. We sometimes retain the older data in a separate ArchiveSets (discussed later).

2.1. Identifiers

Artifact identification is a very broad topic, and we began to address some of its issues elsewhere [2], assigning URIs to each artifact in the system. For this example, we will use some simple tuples comprised of key distinguishing elements.

2.1.1. Granules

Each granule is identified by its type (L0, L1B, L2, L3), the GranuleKey (L0: YYYY-MM-DDTHH[2], L1B/L2: OrbitNumber and L3: YYYY-MM-DD) and an instance identifier. The instance identifier is used to distinguish multiple instances of the same granule, either made in an identical manner, or made from different inputs such as a later version of an algorithm. Since we really want a globally unique instance identifier so that different scientists can independently reproduce granules and distinguish them appropriately, it makes sense for this portion to be a UUID, but they are awkward for humans to work with, so for this paper, we'll use small integers to distinguish instances of the same granule.

2.1.2. Software

We refer to the software encapsulating the integrated algorithms that are used by the system to perform processing as Algorithm Plugin Packages (APPs).

Software versions are relatively straight forward. When a developer makes a change to the source code for a software, it gets delivered into a configuration management system and tagged with a specific version. A rigorous CM process ensures that every change, however minor, gets assigned a distinct version.

The executable software itself is an additional "input" to the process that performs data processing. That executable has its own "Build Process" that produced it from source code, compilers, libraries, and taking place on a particular host with a particular environment. Each of those components are inputs to the process producing the executable. Just as certain elements of the provenance of the data processing are essential for reproducibility of

[2]ISO 8601 date/time `http://www.iso.org/iso/catalogue_detail?csnumber=40874`

scientifically equivalent data, certain elements of the provenance of the software executable (for example, a certain compiler, or a certain version of a library) are essential for reproducing an executable capable of producing scientifically equivalent data files. Other provenance elements are interesting and useful for other purposes, but not strictly essential, like the time it was compiled, or the agent responsible for compiling it.

Again, the dependence on certain aspects of the environment limits reproducibility, while portability and other good software engineering practices can increase reproducibility.

For now, we are flowing the version of the source software through to the executable and suggesting that the versions must match to maintain scientific equivalence. This is clearly not always true. There are many cosmetic changes to the source that don't affect the data contents in a way that would cause them not to be equivalent. It is possible to separate software "Version" into two parts, one indicating a change that affects the data content, and one that doesn't. To date, the utility of such a separation hasn't been worth the effort.

2.1.3. Other Identifiers

Simplifying the identification of lookup tables and calibration, we'll identify them with a name and version as well.

Each of the artifact and process identifiers can be mapped into a URI namespace for constructing RDF triples and asserting facts about those artifacts. We find URIs more useful than literal strings for all artifact identifiers. Each URI can be resolved by the web browser to provide metadata about the artifact.

2.2. Data Aggregation

2.2.1. ArchiveSets

We group granules into **ArchiveSet**s which are processed together. This grouping is arbitrary, but typically corresponds with specific experiments, processing campaigns, etc. This is useful for e-Science since each ArchiveSet can be assigned to an owner who can manage and control processing and data within that ArchiveSet, and grant permissions and sharing of that information. URIs can locate the artifacts associated with each ArchiveSet and allow that data to be accessed remotely or used for further processing. A key concept to an ArchiveSet is that it can never hold two files with the same type and GranuleKey, so {`ArchiveSet,GranuleType,GranuleKey`} is unique at a point in time, or more generally {`ArchiveSet,GranuleType,GranuleKey,Timestamp`} always refers to at most one physical granule, which itself has a globally unique Granule Identifier.

2.2.2. DataSets

Provenance information is tracked down to the granule level during all data processing, but since there are over 5000 orbital granules per year we've found it useful to refer to all the Granules of a given type in the same ArchiveSet as a **DataSet**. We can then summarize some provenance facts at a higher level.

For example, consider production of the first 1000 granules of L2TO3 with version 1.7 of the algorithm in ArchiveSet 1, we can say:
{`L2TO3,1,1`} → {`APP-L2TO3,1.7`}
{`L2TO3,2,1`} → {`APP-L2TO3,1.7`}
{`L2TO3,3,1`} → {`APP-L2TO3,1.7`}, etc.

Those granules are all part of a DataSet {`L2TO3,1`} (type = L2TO3 and ArchiveSet = 1).

Those facts can be summarized and expressed for human consumption by simply saying:
{`L2TO3,1-1000,1`} → {`APP-L2TO3,1.7`}, or as a triple asserting the DataSet {`L2TO3,1`} → {`APP-L2TO3,1.7`}.

If the *L2TO3* APP was upgraded from version 1.7 to version 1.8 at orbit 500, the GranuleKey ranges can be summarized by coalescing ranges:
{`L2TO3,1-500,1`} → {`APP-L2TO3,1.7`},
{`L2TO3,501-1000,1`} → {`APP-L2TO3,1.8`}
or as triples associating both versions of the APP directly with the DataSet:
DataSet {`L2TO3,1`} → {`APP-L2TO3,1.7`},
DataSet {`L2TO3,1`} → {`APP-L2TO3,1.8`}.

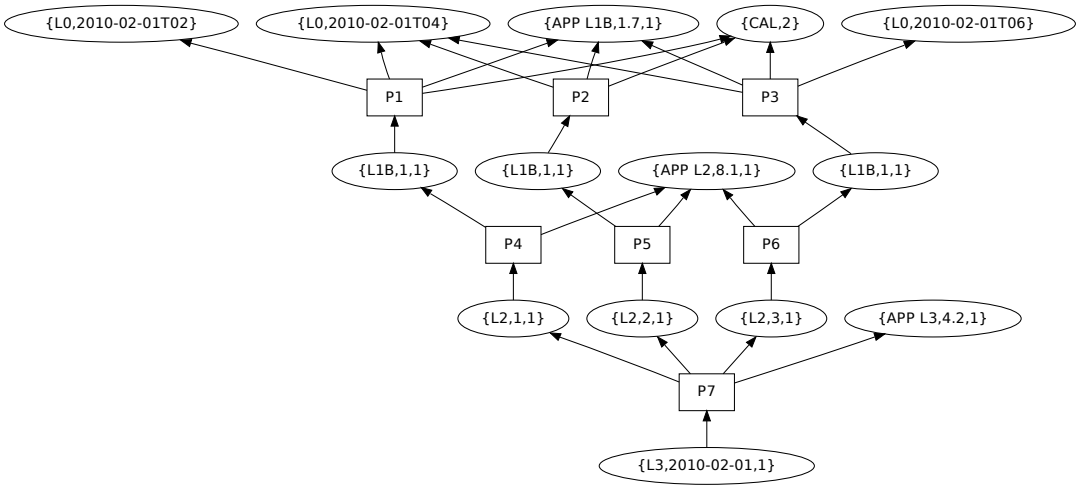

Figure 2: Ozone Example

2.2.3. Versions

As discussed above, the software version is relatively straight forward. Data versions, however, are more complicated. Historically, the most common practice was to simply use the version of the software that produced the data product. Consider the relatively common case of the calibration table, which is an input to the L1B process, changing. Even though the version of the L2 or L3 software hasn't changed, the data files in the whole process have been affected by the change in the calibration.

ArchiveSets allow us to separate, identify, and refer to the DataSets that have been produced in different manners, not just in terms of the version of the immediate predecessor software, but isolating a whole set of changes from one another.

As granules are added to an ArchiveSet, they become part of the DataSet for their data type. Within the ArchiveSet, we also maintain identifiers for the data processes in the workflow. ^{my}Grid uses a similar scheme for establishing identities of workflow provenance and data products. [6] In particular, the Taverna IDSet aggregation identity scheme has similarities to our DataSet concept, though ArchiveSets and GranuleKey geospatial/temporal enumerations simplify aggregation for the special case of earth science data. We follow the *Strong Identification* and via our DataSets, the *Strong Identification with IDSet* identification strategies as described by Chapman and Jagadish. [7]

As noted, this process is operational, so new granules are added every day. In order to precisely refer to a specific set of granules, we timestamp each granule on ingest, and allow a date/time differentiator on a DataSet granule membership query to determine the specific list of granules.

3. Provenance Equivalence Identification

A more specific example of the data processing graph is shown in figure 2. This graph uses the historical provenance convention, where the arrows indicate the USED and WASGENERATEDFROM relationships, so are backward from the data flow diagram. The figure shows the basic input files, and a link to the software used in each process, but omits much other information, including the provenance graph of the software, and the compute environment.

3.1. Single Level

The first case is to reproduce the final granule, {`L3,2010-02-01,1`}. A query of the identifier for that granule will discover that it was produced from {`L2,1,1`}, {`L2,2,1`} and {`L2,3,1`} using the software {`APP L3,4.2,1`}.

If they are all available, we can simply re-run the APP and produce the output granule.

3.2. Multi Level

Assume the immediate predecessor files are not available. The science team has delivered a new version of the Level 2 lookup table, {LUT,8 }, and all of the L2 data have been reprocessed. Instead of those granules, the system holds {L2,1..3,2 }.

Querying the provenance of the original files shows how they were produced, and links to the specific input granules from them. So, for example, {L2,1,1 } was produced from {L1B,1,1 } and {LUT,7 } by {APP L2,8.1,1 }. The current granule however is {L2,1,2 }, produced from tupleL1B,1,1 and {LUT,8 } by {APP L2,8.1,1 }. Since the updated LUT causes the resulting granule not to be scientifically equivalent, it can't be used by our reproducing process.

We must instead additionally reproduce {L2,1,1 }. Since all of its inputs are available, it can be re-run and will produce a new granule {L2,1,3 }. This newly produced granule is distinct and has a distinct identifier from the original{L2,1,1 }. The associated provenance information about when it was produced, where it was produced, etc. are all different and are linked individually (represented by semantic tagging) to the distinct identifiers. The important fact though is that the distinct granules {L2,1,1 } and {L2,1,3 } have the same *essential provenance* and assuming the process is *scientifically reproducible*, should be *scientifically equivalent* to on another. Also, each of them are not *scientifically equivalent* to the other granule, {L2,1,2 }.

3.3. Provenance Equivalence Identifiers

Starting from the root of the graph, we can tag each granule with an additional identifier to represent the provenance equivalence of that granule. We construct it by computing a digital signature of the elements of *Essential Provenance* of the process that produced that granule. For the raw, level 0 data, we use a hash of our unique identifier for that granule.

$$\begin{aligned}\{\texttt{L1B,1,1}\} \rightarrow H(&\{\texttt{L0,2010-02-01T02,1}\},\\ &\{\texttt{L0,2010-02-01T04,1}\},\\ &\{\texttt{CAL,2}\},\\ &\{\texttt{APP L1B,1.7,1}\})\\ \{\texttt{L1B,1,1}\} \rightarrow\ & h_1\end{aligned}$$

$$\begin{aligned}\{\texttt{L2,1,1}\} \rightarrow H(&h_1,\\ &\{\texttt{LUT,7}\},\\ &\{\texttt{APP L2,8.1,1}\})\\ \{\texttt{L2,1,1}\} \rightarrow\ & h_2\end{aligned}$$

The newly processed {L2,1,2 } was made with a different lookup table, so it has a different digital signature:

$$\begin{aligned}\{\texttt{L2,1,2}\} \rightarrow H(&h_1,\\ &\{\texttt{LUT,8}\},\\ &\{\texttt{APP L2,8.1,1}\})\\ \{\texttt{L2,1,2}\} \rightarrow\ & h_3\end{aligned}$$

but since we reproduced {L2,1,3 } in a scientifically equivalent manner, it gets the same signature:

$$\{\texttt{L2,1,3}\} \rightarrow H(h_1, \{\texttt{LUT,7}\}, \{\texttt{APP L2,8.1,1}\})$$
$$\{\texttt{L2,1,3}\} \rightarrow h_2$$

Now to reproduce {L3,2010-02-01,1 }, we look up its immediate predecessors, and find that the specific input granule was {L2,1,1 }, which isn't available, but we can request other granules with the same Provenance Equivalence Identifier (PEI) that should be scientifically equivalent to the one we desire, since they have equivalence hash h_2, and find that granule {L2,1,3 } is an acceptable (scientifically equivalent) granule to use.

3.4. Scientifically Equivalent Software

This technique can also be applied to the software itself to represent scientically equivalent software. We include in the essential provenance of the software executable all of the dependencies both on the compiler versions and the library versions that can affect scientific equivalence. In this way, an executable is tagged not just with a specific version, but also with a Provenance Equivalence Identifier summarizing the manner in which it was built. As discussed above, the versions of some of of those elements could (through analysis or testing validation) determined to be scientifically equivalent and tagged as such. For example, if it was determined (and validated) that libx version 1.8.1 was scientifically equivalent to libx version 1.8.2 (even though they both might be different from version 1.7.1), those two versions could be tagged with the same signature by excluding that minor version.

4. Semantic Provenance Tagging

4.1. Data Citations

A major purpose for data set identifiers is for data citations. The particular contents of proper data citations is beyond the scope of this paper [8], but are part of the use case of concern. If a reader of a journal article citing a dataset wants to reproduce that dataset in order to reproduce the science described by the research, the citation, and the data identifier must be sufficient to produce a similar data set that is Scientifically Equivalent to the cited dataset. This requires an identifier that can be resolved to the level of granule membership, software source version, and sufficient identification of the software build environment and compute environment for an independent party to reproduce the data set.

Consider the problem of Data Citation comparison. Two researchers access similar, but not identical dataset. They each rigorously cite their data, but those data identifiers each refer to very large sets of data. The question then comes, what is the difference? Each of the dataset identifiers provide entry into potentially extremely large semantic web graphs of their entire granule membership and provenance graphs for each of those granules. A mechanical process of comparing those provenance graphs becomes very difficult and time consuming. Categorizing certain provenance properties as essential for reproducibility, and tagging each granule with a Provenance Equivalence Identifier summarizing the provenance can make that comparison process easier.

4.2. Data Model and Ontologies

Our system assigns a unique, persistent URL [3] to each artifact in the system, then relates those artifacts to one another and represents their relationship and using various ontologies. The basic workflow information maps into the Open Provenance Model (OPM) [9], and can be exported in the XML or semantic web RDF representations of that data model. We are using some other standard ontologies for computer inventory information and related context information and some local experimental ontologies for tagging our application specific information, including the Provenance Equivalence Identifier.

[3] http://purl.org

5. Conclusions and Future Work

Complete provenance capture can enable complete reproducibility of a complex process by going back to the raw data. Systematic semantic tagging of provenance equivalence based on cascaded digital signatures of a canonical collection of essential provenance elements can enable subsets of the process to be sampled and reproduced reliably at a higher level. When data sets are different, Provenance Equivalence Identifiers can also simplify comparison of data set provenance graphs and help summarize the differences.

We plan on implementing more of the increasingly standardized provenance ontologies for the representation of provenance information from our data processing system. We will document a standard essential provenance canonicalization representation so the digital signatures can be produced by others reproducing our process. Our goal is to extend our dataset and granule identifiers and complete provenance artifacts into the linked data cloud and provide a service for comparing datasets through their provenance graphs.

Acknowledgment

The authors would like to acknowledge contributions from the ACPS software development team and numerous discussions with the NASA Earth Science Data Systems Working Group (ESDSWG[4]) and the Federation of Earth Science Information Partners (ESIP FED[5]) Data Preservation Cluster.

References

[1] C. Tilmes, A. J. Fleig, Provenance Tracking in an Earth Science Data Processing System, in: Provenance and Annotation of Data and Processes, Vol. 5272 of Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 2008, pp. 221–228. doi:10.1007/978-3-540-89965-5_23.
URL http://www.springerlink.com/content/3433g6633h91781k/

[2] C. Tilmes, Y. Yesha, M. Halem, Provenance Artifact Identification in the Atmospheric Composition Processing System (ACPS), Proceedings of the 2nd Workshop on the Theory and Practice of Provenance.
URL http://www.usenix.org/events/tapp10/tech/full_papers/tilmes.pdf

[3] B. Barkstrom, A mathematical framework for earth science data provenance tracing, Earth Science Informaticsdoi:10.1007/s12145-010-0057-0.
URL http://dx.doi.org/10.1007/s12145-010-0057-0

[4] P. Bhartia, Total ozone from backscattered ultraviolet measurements, in: G. Visconti, P. Carlo, W. Brune, A. Wahner, M. Schoeberl (Eds.), Observing Systems for Atmospheric Composition, Springer New York, 2007, pp. 48–63, 10.1007/978-0-387-35848-2_3.
URL http://dx.doi.org/10.1007/978-0-387-35848-2_3

[5] C. Tilmes, Y. Yesha, M. Halem, Tracking provenance of earth science data, Earth Science Informatics 3 (2010) 59–65, 10.1007/s12145-010-0046-3.
URL http://dx.doi.org/10.1007/s12145-010-0046-3

[6] C. G. Jun Zhao, R. Stevens, in: Provenance and Annotation of Data, Vol. 4145 of Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 2006, pp. 254–269. doi:10.1007/11890850, [link].
URL http://www.springerlink.com/content/ml4113ln11r538u0/

[7] A. Chapman, H. V. Jagadish, Provenance and the Price of Identity, Springer-Verlag, Berlin, Heidelberg, 2008, pp. 106–119. doi:10.1007/978-3-540-89965-5_12.
URL http://portal.acm.org/citation.cfm?id=1484346.1484360

[8] M. Parsons, R. Duerr, J.-B. Minster, Data citation and peer review, EOS Vol. 91 (Num. 34).

[9] L. Moreau, B. Clifford, J. Freire, J. Futrelle, Y. Gil, P. Groth, N. Kwasnikowska, S. Miles, P. Missier, J. Myers, B. Plale, Y. Simmhan, E. Stephan, J. V. den Bussche, The open provenance model core specification (v1.1), Future Generation Computer Systems.
URL http://eprints.ecs.soton.ac.uk/21449/

[4] http://esdswg.eosdis.nasa.gov/

[5] http://www.esipfed.org/