Preserving Reproducibility: Provenance and Executable Containers in DataONE Data Packages

Bryce Mecum (mecum@nceas.ucsb.edu; https://orcid.org/0000-0002-0381-3766)
Matthew B. Jones (jones@nceas.ucsb.edu; https://orcid.org/0000-0003-0077-4738)
Dave Vieglais (dave.vieglais@gmail.com; https://orcid.org/0000-0002-6513-4996)
Craig Willis (willis8@illinois.edu; https://orcid.org/0000-0002-6148-7196)

Many formats and standards for packaging research objects exist, each with communities of practice, nomenclature, intended use cases, and target audiences. All of these formats share the common goal of grouping together the digital output of scientific research into a composite container: a Research Object. DataONE refers to these composite Research Objects as Data Packages, and contain all data, metadata, software, and other products of research like figures and graphs (Figure 1). In addition to merely packaging together input and output objects from research, some packaging formats also provide additional features such as rich metadata using XML schemas, provenance, and serialization formats. Choosing a particular packaging format is a challenging task for both researchers and repository operators alike.

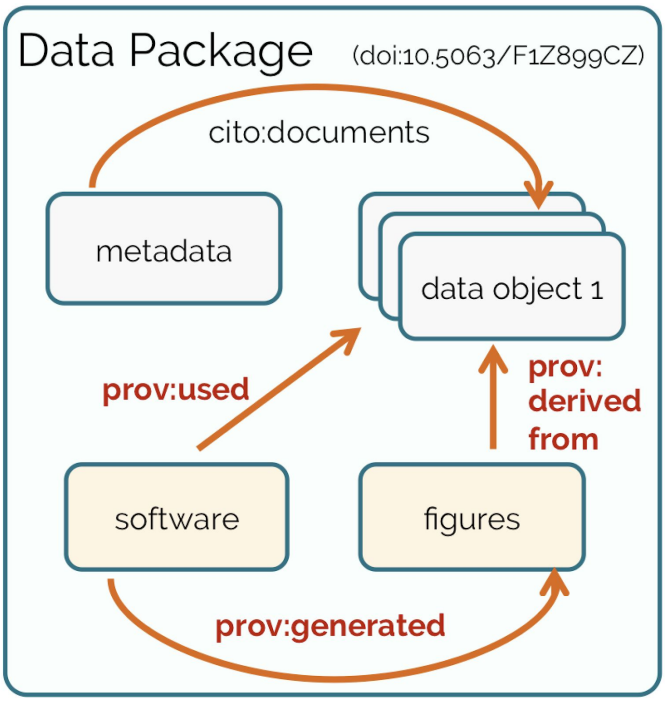

Figure 1: A DataONE Data Package is a composite container for data, software, and visualizations, and the metadata that describes these. Provenance relationships link the objects within and across packages into computational workflows for reproducible science.

DataONE is a federation of data repositories that federates data and metadata from over 45 data repository systems. DataONE provides packaging for composite research outputs via the commonly-used Open Archives Initiative Object Reuse and Exchange (OAI-ORE) (Lagoze et al. 2012) Resource Map standard, and supports optional serialization using BagIt (Kunze et al. 2018). At DataONE's inception, it was deemed important to make use of established and open standards where practical. OAI-ORE provides an open mechanism for describing aggregations of distinct resources on the web by using their respective URIs to provide linkages in an Resource Description Framework (RDF) model. OAI-ORE uses RDF predicates to provide relationships between aggregated components, enabling tremendous flexibility for describing package constructs and any

annotations therein. DataONE's Data Package [1] specification is a convention of using OAI-ORE with a set of additional constraints on top of the ORE standard to improve preservation, access, and linking of data packages:

- When serializing a package in BagIt format, the ORE Resource Map must be provided in RDF/XML format with a particular filename in the BagIt bag.
- Each resource with a representation in a DataONE ORE package MUST be described with an *dcterms:identifier* containing the DataONE persistent identifier.
- All DataONE Objects in the map MUST be expressed as a URI using DataONE's resolving service, instead of using a URI to a specific replica on a repository. This is to separate the current physical location of a resource from its identity.
- A mapping between the dcterms:identifier and the file location in the BagIt data directory must be provided in a manifest file named ' pid-mapping.txt'. This allows a direct correspondence to be discovered between local objects in a serialized bag and remote resource URIs in the ORE document by using the persistent identifier to link them.
- The aggregation resource URI SHOULD be expressed as a hash URI based on the resource map URI, as recommended by ORE (see: http://www.openarchives.org/ore/1.0/primer#remHashURIs, and http://www.openarchives.org/ore/1.0/http#Simple) This ensures that the aggregation can be referenced directly in other Resource Maps and still be resolved.
- When referencing another DataONE Data Package, the URI of the Data Package being referenced MUST resolve to a Resource Map. The URI can either be the Resource Map URI or the aggregation URI if it follows the hash URI format. Since some existing Resource Maps do not use aggregation URI's that resolve to the Resource Map, it is necessary to check their format before deciding which to use.
- When expressing an identifier in a URI, it must be URL encoded. When expressing in the dcterms:identifier field, it must not (although appropriate XML encoding applies).
- The Resource Map MUST assert a statement with the ore:isDescribedBy relationship between the Resource Map and the aggregation, following the recommendation that aggregations with multiple resource maps express this relationship (see http://www.openarchives.org/ore/1.0/datamodel#ReM-to-aggr).

These rules, while minor, allow DataONE to successfully federate a collection of research result objects and their associated metadata as a DataONE Data Package and provide effective search and discovery tools for researchers. The decision to re-use an existing standard had multiple advantages: (1) Data Packages can be parsed and serialized by existing, well-tested software tools, which has saved DataONE time, and (2) Data Packages have meaning outside of DataONE's Data Package standard in that they can be treated as ORE Resource Maps and are therefore interoperable with similar systems.

Incorporating Provenance in Packages

Archiving the input and output objects of research for later access is a key piece of a reproducible scientific process. However, without information about how those research objects came into existence and relate to one another, the research is likely not reproducible by another scientist. For example, one needs to know which computational processes used which input objects, which software was used to drive the computation, and which output objects were produced, often in a complex workflow consisting of hundreds of steps.

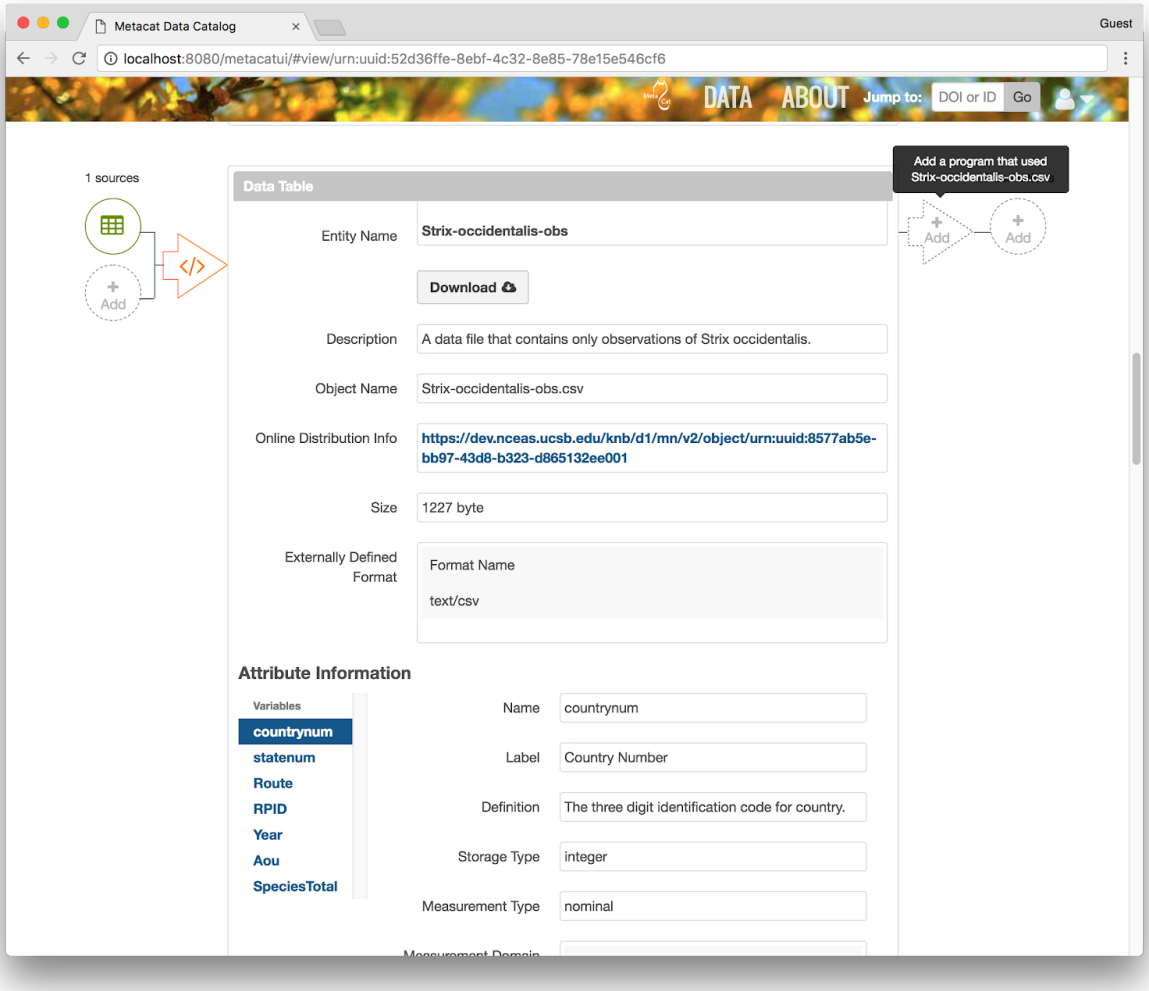

Figure 2: Provenance information can be viewed and even authored directly from a dataset's landing page using DataONE's provenance editor. Provenance is shown using a workflow metaphor using icons for source objects, processing scripts, and derived objects.

The solution DataONE has taken is to include structured provenance metadata in datasets as part of the enclosing Data Package. To do this, DataONE created ProvONE [2], a Web

Ontology Language (OWL) ontology that extends the W3C PROV [3] standard for describing the provenance of computational workflows. ProvONE represents provenance information in the form of RDF/XML that is inserted in the Resource Maps that define Data Packages. Adding provenance to a data package in this way is straightforward because ORE Resource Maps already are modeled with RDF and therefore can include other arbitrary models such as ProvONE.

To assist scientists in creating and consuming this provenance information, DataONE provides a rich web display (Figure 2) on Data Package landing pages and two packages for the R Programming Language (R Core Team 2018): recordr [4] for automatically recording provenance during R sessions, and datapack [5], for serializing provenance information into Data Packages.

Together, the Data Package model and ProvONE model, along with accompanying support in user-facing software tools, enable researchers to effectively describe the content and provenance of their research products.

Implementation considerations

While making use of the existing OAI-ORE standard for packaging in DataONE has had advantages, the choice was not without the need for careful implementation. Designing a new packaging standard might have been easier due to not having to consider interoperability with other communities and being able to optimize the standard against technology stacks for performance reasons. However, Resource Maps have been well-suited to the needs of the Data Package standard with only minor implementation considerations needed along the way.

The first consideration has been in building effective search and discovery interfaces based upon Resource Maps. Because Resource Maps support the full semantic and logical richness of RDF and OWL, it was tempting to to make the search indexes behind DataONE's search interfaces support that same level of richness. Existing tools for processing Resource Maps [6] work very well for smaller packages but exhibit exponential increases in processing time and computing resource usage for larger (> 1000 object) packages. To work around these performance issues, some member repositories in DataONE have artificially limited the number of objects that can be included in a Data Package, even though science is often done at a scale beyond 1000 objects.

The second implementation consideration is dealing with the inherently flat nature of Resource Maps. With Resource Maps, all objects are described at the same level of hierarchy (in an Aggregation). However, when present on the creating scientist's filesystem, objects are often arranged in a hierarchy of files and folders that confers rich semantic relationships among the objects and their use in the research. Numerous repositories approach this limitation by making use of nested Resource Maps to match the nested structure of the filesystem, where each level

of hierarchy is described by a separate Resource Map that can contain other Resource Maps representing child folders. This worked, but was made more difficult to implement in DataONE for two reasons. First, because all DataONE objects are immutable and have their own unique identifiers, adding or changing the identifier of a child Resource Map requires an update all parent Resource Maps up to the topmost parent inthe Data Package hierarchy, which is computationally intensive and requires careful implementation in software tools. Second, because the Data Package specification requires each Data Package to have at least one metadata record aggregated within it, metadata records throughout complex hierarchies tended to either be too minimal to support standalone interpretation and/or contained highly-redundant information which made it hard for users to discern between Data Packages in search interfaces.

The third implementation consideration is less specific to OAI-ORE and more due to the interaction between OAI-ORE and the DataONE Object model (which Data Packages aggregate). In DataONE, any changes in the content of an object requires a new identifier for the object, and, thus, building user-facing tools that do both the right thing and won't surprise users requires careful consideration. For example, if a user authors a Data Package with a metadata record describing an Excel spreadsheet and they decide to replace the Excel spreadsheet with a Comma-Separated Values (CSV) version of the same data, do the references (e.g., that it was derived from it) in the Resource Map (which were asserted by identifier) still apply to the CSV version of the data? Should the triples that referenced the Excel spreadsheet be automatically removed for the user? Or if the user has provenance information embedded in their Resource Map and they author provenance that details an R script that generated a figure, what happens in the Resource Map if they update the R script (resulting in a new DataONE Object). Should the triple connecting the previous version of the R script be removed or updated to connect the new version? Should both be retained? These kinds of problems are tractable, but need careful attention in user interfaces to make it clear to the user what changes their actions are going to cause and to provide sensible default or worst-case behavior.

Comparison with other Research Object packaging standards

Of the myriad package standards available for use today, there are two camps: Those that use Resource Maps and BagIt and those that do not. Standards in the Resource Map + BagIt group include RDA Repository Interoperability Package (RDA Research Data Repository Interoperability Working Group 2018), Research Objects [7], DataONE Data Packages [8], and Data Conservancy Packages [9]. Packaging standards in the other camp, mostly provide for similar functionality but use different technologies. For example, the Frictionless Data datapackage.json [10] makes use of JSON and the DataCrate [11] uses JSON-LD. Despite these differences in serialization, there are few fundamental differences between the formats that can't be addressed by converting from one serialization to another, and so the research infrastructure community would benefit from consolidation on a shared standard like the RDA

Repository Interoperability Package (RDA Research Data Repository Interoperability Working Group 2018).

Representing executable containers as packages

The WholeTale project[1] aims to improve reproducibility in computational research by providing a platform and collaborative environment for the creation of standards-based composite research objects referred to as "Tales". Tales include descriptions of the computational environment, and the code, metadata, data objects (or references), and other inputs and outputs needed to fully reproduce a computational result (Brinckman et al. 2018). The Whole Tale platform is designed to support the creation, validation, and execution of Tales as well as publication to external repositories, including DataONE member repositories.

In our view, Tales are just extensions of the concept of a DataONE Data Package to include the additional metadata and objects needed to fully reconstruct and re-execute a computational workflow that produced a result. Therefore, WholeTale provides first-class publication of Tales via DataONE Data Packages and serialization outside of DataONE via BagIt. Tales are composed of data, code, any output, as well as metadata, provenance, and, crucially, a portable description of the computation environment (e.g., Dockerfile plus any supporting files) under which the output was produced so that another researcher can reproduce them.

To support exchange of Tales outside of DataONE, WholeTale will make use of the Research Data Alliance's (RDA) Research Data Repository Interoperability (RDRI) standard (RDA 2018) which uses a BagIt serialization. RDRI focuses on storing metadata alongside data in a canonical location. The Tale model extends this, adding information about the execution environment environment in the form of Dockerfiles and additional provenance stored in an associated Resource Map.

Tales can be compared to related initiatives such as CodeOcean capsules (CodeOcean, 2018), the Opening Reproducible Research (O2R) initiative's Executable Research Compendium (ERC) packages (Nüst et al, 2017) and smart containers (Huo, 2015). Like CodeOcean, Whole Tale provides a collaborative platform for the creation of reproducible computational research. CodeOcean will soon support exporting capsules[2], also based around Docker, using an internally-defined YAML format. As an open-source platform, Whole Tale is designed around standards-based formats to ensure that Tales are shareable and re-runnable outside of the Whole Tale service. In addition to capsules, the nascent ERC specification suggests further opportunity for standardization and interoperability around these and related formats.

[1] https://www.wholetale.org
[2] Private communication. 7/15/2018

The future of Data Package

The DataONE Data Package standard has served DataONE well since its inception by providing the federation with a standards-based packaging approach that enables DataONE to quickly onboard member repositories into the federation, increasing the usefulness of the DataONE federation as a whole. A missing feature of the Data Package standard, as outlined above, has been the lack of a mechanism for packaging objects hierarchically within a single Data Package. Researchers often make extensive use of hierarchical filesystems when organizing filesystem objects, and these filesystem hierarchies are often important norms within their respective communities. To support this use case, Data Package is being extended to support optional annotations of filesystem paths on each resource in the Resource Map, thus allowing the file hierarchies to be reconstructed. The Research Object “ro” vocabulary [13] is being considered as an implementation approach for this feature.

Also on the horizon for Data Package is adding support for alternative serialization formats, including JSON-LD. JSON-LD is becoming increasingly popular on the web and offers a number of advantages over RDF/XML while still providing the semantic richness of the RDF model. JSON-LD is considered by some to be more human-readable than RDF/XML and, despite the best efforts of repositories, humans eventually end up needing to read packaging formats. Also, an increasing number of software stacks are built on top of the web, which is now largely unified around JSON rather than XML. Moving to a JSON-LD serialization for our ORE Resource Maps, which would be valid JSON, will allow DataONE member repositories to take advantage of this shift in technology.

Summary

Making use of open standards wherever possible has served DataONE well since its inception. Choosing OAI/ORE Resource Maps and BagIt as standards to build on top of has allowed the DataONE Data Package to be easily implemented across a network of repositories and extended over the years as features such as provenance were added to the standard and the ecosystem at large. Careful implementation was needed across technology stacks making use of the Data Package standard, but use of an open standard built on top of rich and established technologies such as RDF has allowed DataONE to extend Data Package to support new use cases, and it continues to allow Data Package support for new features such as describing object hierarchies and executable packages in the form of Whole Tales “Tales”.

Footnotes/Links

[1] https://releases.dataone.org/online/api-documentation-v2.0/design/DataPackage.html
[2] https://purl.dataone.org/provone-v1-dev
[3] https://www.w3.org/TR/prov-overview/

[4] https://github.com/NCEAS/recordr
[5] https://github.com/ropensci/datapack
[6] https://github.com/abrin/foresite-toolkit
[7] http://www.researchobject.org/
[8] https://releases.dataone.org/online/api-documentation-v2.0/design/DataPackage.html
[9] http://dataconservancy.github.io/dc-packaging-spec/
[10] https://frictionlessdata.io/specs/data-package/
[11] https://github.com/UTS-eResearch/datacrate
[12] http://wholetale.org/
[13] http://wf4ever.github.io/ro/2016-01-28/ro/

References

Brinckman, A., Chard, K., Gaffney, N., Hategan, M., Jones, M. B., Kowalik, K., … Turner, K. (2018). Computing environments for reproducibility : Capturing the “Whole Tale.” Future Generation Computer Systems. http://doi.org/10.1016/j.future.2017.12.029

CodeOcean. (2018). "What is a compute capsule?" Downloaded from https://help.codeocean.com/getting-started/what-is-a-compute-capsule

Carl, L., de Sompel Herbert, V., Michael, N., Simeon, W., Robert, S., & Pete, J. (n.d.). A Web-based resource model for scholarship 2.0: object reuse & exchange. Concurrency and Computation: Practice and Experience, 24(18), 2221–2240. http://doi.org/10.1002/cpe.1594

Kunze, J., J. Scancella, C. Adams, L. Madden, and J. Littman. (2018). The BagIt File Packaging Format (V1.0). https://tools.ietf.org/html/draft-kunze-bagit-16.

Huo, D., Nabrzyski, J., & Vardeman II, C. F. (2015). An Ontology Design Pattern towards Preservation of Computational Experiments. In LISC@ ISWC (pp. 15-18).

Lagoze, C. , Van de Sompel, H. , Nelson, M. , Warner, S. , Sanderson, R. and Johnston, P. (2012), A Web-based resource model for scholarship 2.0: object reuse & exchange. Concurrency Computat.: Pract. Exper., 24: 2221-2240. doi:10.1002/cpe.1594
Marwick, B., Boettiger, C., Mullen, L., Marwick, B., Boettiger, C., & Mullen, L. (2018). Packaging Data Analytical Work Reproducibly Using R (and Friends) Packaging Data Analytical Work Reproducibly Using R (and Friends) ABSTRACT. The American Statistician, 72(1), 80–88. http://doi.org/10.1080/00031305.2017.1375986

Nüst, D., Konkol, M., Schutzeichel, M., Pebesma, E., Kray, C., Przibytzin, H., & Lorenz, J. (2017). Opening the publication process with executable research compendia. D-Lib Magazine, 23(1/2).

R Core Team (2018). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

RDA Research Data Repository Interoperability Working Group. 2018. Research Data Repository Interoperability WG Final Recommendations. URL http://dx.doi.org/10.15497/RDA00025.