Automating Data Curation Processes

Reagan W. Moore
University of North Carolina at Chapel Hill
216 Manning Hall
Chapel Hill, NC 27599-3360
01 919 962 9548
rwmoore@renci.org

ABSTRACT

Scientific data quality can be abstracted as assertions about the properties of a collection of data sets. In this paper, standard properties are defined for data sets, along with the policies and procedures that are needed to enforce the properties. The assertions about the collection are verified through periodic assessments, which are also implemented as policies and procedures. Data quality curation can then be defined as the set of policies and procedures that verify the scientific data quality assertions. The assertions made by the creators of a collection are differentiated from the assertions about data quality needed by the users of the data. The transformation of data into a useable form requires well-defined procedures that need to be considered as part of the data curation process. The automated application of both digital curation and data transformation procedures is essential for the management of large scientific data collections.

Categories and Subject Descriptors

D.4.7 [**Operating Systems**]: Organization and Design – *distributed systems*

General Terms

Management, Design, Verification.

Keywords

Policy-based data management.

1. Data quality

Scientific data quality is dependent on the specification of a scientific research context. The creators of a scientific data set are typically driven by a research question, and choose quality criteria that are necessary for exploration of a research issue. The criteria may include properties that each data set must possess (such as physical units), or properties that are related to the entire collection (such as completeness and coverage). The properties can be turned into assertions that the data set creators make about their collection. Scientific data quality is quantified by the collection creators by explicitly verifying compliance with the desired properties.

The types of properties that are associated with scientific data sets can be loosely categorized as:

- Data format (e.g. HDF5, NetCDF, FITS, …)
- Coordinate system (spatial and temporal locations)
- Geometry (rectilinear, spherical, flux-based, …)
- Physical variables (density, temperature, pressure)
- Physical units (cgs, mks, …)
- Accuracy (number of significant digits)
- Provenance (generation steps, calibration steps)
- Physical approximations (incompressible, adiabatic, …)
- Semantics (domain knowledge for term relationships)

Additional properties can be derived from these categories. Thus the relevant time period may be defined in the temporal coordinate, and allowed transformations may be implicit in the type of variables and physical approximations that were made in creating the data collection. The additional properties may be evaluated by applying procedures that generate the desired information, which in turn can be applied to the data sets as metadata.

Data curation corresponds to identifying the properties claimed by the data set creators, verifying that the desired properties are consistently present throughout the data set, logging any discrepancies, and assembling an archival information package. Each desired property requires the evaluation of knowledge to determine its presence, and the creation of information that is associated with each data set as metadata. By examining the metadata, a user of the collection can determine whether the properties are present that are needed for the user's research initiative. Data quality from the user's perspective is determined by compliance with the properties that are needed when incorporating a data set into the user's analysis environment or data collection.

Data quality is a mapping from assertions that the creators of a collection make about their data sets, to requirements by a user for the appropriateness of the data sets for use in their own research. Both persons may have equally valid but incommensurate criteria for data quality.

2. Data, Information, and Knowledge

Data curation can be thought of as an active process that requires assessment of the information and knowledge context associated with a collection. The OAIS model uses the terms "representation information" and "knowledge community" to express the requirement that a targeted community be able to understand and manipulate a collection based on the representation information. In order to automate the generation of representation information,

Curating for Quality Workshop, September 11-12, Arlington, VA.

we need a definition of representation information that is computer actionable. We define:

- Data - consists of bits (zeros and ones)
- Information - consists of labels applied to data
- Knowledge - defines relationships between labels
- Wisdom - defines when and where knowledge relationships should be applied (relationship on relationships)

If we have a set of computer actionable processes that apply representation information, we can automate data curation actions. Since scientific data collections may be comprised of hundreds of millions of files and contain petabytes of data, automation is essential for building a viable preservation environment.

Policy-based data management systems provide computer actionable mechanisms for defining information, applying knowledge, and governing policy execution.

- Information is treated as labels that are applied to data sets as metadata. Each data set may have both system defined and user defined metadata attributes that are persistently maintained. System metadata consists of pieces of information that are generated when processes are applied to data sets. An example is a process that creates a replica. The location of the replica is system metadata that is associated with the data set.
- Knowledge is treated as procedures that evaluate relationships. While information is treated as static metadata that is maintained persistently, knowledge is treated as an active process that involves the execution of a procedural workflow. To simplify creation of knowledge procedures, basic functions called micro-services are provided that encapsulate well-defined actions. The micro-services can be chained together into a workflow that is executed whenever the associated knowledge is required.
- Wisdom is applied through policy enforcement points that determine when and where knowledge relationships should be evaluated. Each external action is trapped by a set of policy enforcement points within the data grid middleware. At each policy enforcement point, the data grid checks whether a policy should be applied. The policy enforcement points can control what is done before an action is executed, can control the action itself, and can control what is done after an action takes place. A simple example is the control of what happens when a file is ingested into a collection. The data grid middleware may transform the data set to an acceptable archival format, extract provenance metadata, generate a checksum, and replicate the data set.

This defines the minimum system components that are needed to automate curation processes. Fortunately, policy-based data management systems implement the above mechanisms for managing information, generating knowledge, and applying wisdom.

3. POLICY-BASED DATA MANAGEMENT

The integrated Rule Oriented Data System (iRODS) is used to build data curation environments [1]. The system is sufficiently generic that the iRODS middleware is used to implement all stages of the data life cycle. This approach to data curation is based on the following principles:

- The **purpose** for creating the collection determines the **properties** that should be maintained including data quality.
- The **properties** of the collection determine the **policies** that should be enforced.
- The **policies** control the execution of **procedures** through computer actionable rules.
- The **procedures** apply the required knowledge relationships and generate **state information** through computer executable workflows.
- The **state information** (metadata) is saved persistently.
- **Assessment criter**ia can be evaluated through periodic execution of policies that query the **state information** and verify that re-execution of the procedures generates the same result.

This provides an end-to-end system that enforces the required curation policies, persistently manages the representation information, and enables validation of data quality assessments.

The iRODS data grid provides representation information about the preservation environment through the set of policies and procedures that are applied. This representation information quantifies the data curation policies. The system can be queried to discover which policies are being applied. The procedures can be re-run to verify that the system is maintaining the quality metrics correctly.

A simple example is an integrity criterion that asserts that the data sets have not been corrupted. One approach is to save a checksum that is formed by manipulating all of the bits in the file. If any of the bits have been corrupted, the checksum will change. The original checksum for the file (created at the time of ingestion) can be saved as persistent state information. The policy that governs the creation of the checksum can be re-run at any point in the future, generating a new checksum. The original value and the most recently created value can be compared to verify the integrity of the file.

3.1 Knowledge Scale

An important question is the whether it is feasible to quantify information, knowledge, and wisdom as metadata, policies/procedures, and policy enforcement points. Will the number of entities remain bounded, or will the amount of system representation information become larger than the collection size?

Applications of the iRODS data grid typically maintain:

- About 220 attributes associated with files, users, collections, storage systems, policies, and procedures. Note that system level metadata is needed not only for files, but also for the preservation environment itself.
- About 74 policy enforcement points for controlling the execution of policies. Policy enforcement may be imposed before an action is executed, to control an action, and after an action is executed.
- About 250 micro-services for implementing procedures [2]. Examples include micro-services to query the metadata catalog, loop over the result set, read a file, create a checksum, store new descriptive metadata attributes, replicate a file, etc.

- About 20 rules that enforce collection properties such as data quality. This number typically corresponds to about 20 properties that are desired for a collection. Note that the system is capable of supporting thousands of rules, but most applications choose a small subset. Examples might be rules that maintain integrity (replicate a file, validate checksums), maintain authenticity (manage provenance information), track chain of custody (audit trails), and track original arrangement (physical file path).

When enforcing assertions about data quality, a data management system needs a computer actionable rule that controls the extraction of the desired property, and a computer executable workflow that applies the required relationships.

4. CURATION APPLICATIONS

We can conduct a thought experiment to decide how we can automate data quality assessments about a collection, based on the properties of scientific data collections listed in Section 1. An immediate question is whether a specific data quality property requires the extraction of metadata from within each data set, or whether the information must be provided through an external mechanism. A related question is whether the properties will be uniform throughout the data collection, or whether some properties will be unique to a sub-set of the files. Another possibility is that the desired data quality property has to be determined through examination of the actual data, such as detection of missing values. The types of processing that are applied to verify data quality will vary dramatically based on the type of data, desired properties for a collection, and purpose behind the generation of the data set.

The following examples are intended to demonstrate that data quality inherently is dependent upon the execution of procedures that verify the presence of desired properties either within each data set or within a collection. Data quality curation can be defined in terms of the data quality procedures that are executed by either the creator of a collection or by users of a collection. These procedures may be quite different and result in different definitions of the quality of a data collection.

Quality assessment for **data format** tends to be evaluated for each data set within a collection. Each data type has a standard structure that can be verified. The HDF5, NetCDF and FITS data formats package metadata with the data. The metadata can be extracted and registered as queriable attributes within a collection. Given a standard structure for the data format, micro-services can be created that evaluate the structure, verify the structural components are consistent with the specification, and ensure that the data can be read from the structure in the future. An expectation is that the micro-service that analyzes and manipulates the data structure will be executable on future architectures. A quality assessment procedure that is executed today should also be executable in the future on future operating systems. Data grids provide this capability through virtualization of standard I/O operations.

Quality assessment for the **coordinate system** is typically information that is evaluated for each data set. For gridded data, the spatial and temporal location may be explicitly stored, or may be inferred from spatial dimension arrays. An example of the importance of correctly assigning the coordinate system occurs when satellite data is geo-registered. The quality of the data depends upon the ability to correlate a pixel in a satellite image with a point on the ground. The algorithm that does the correlation is an essential component of a data quality assessment that may need to be reapplied in the future. This particular geo-referencing procedure is typically done by the creator of the data collection.

The **geometry** associated with the coordinate system is rarely explicitly captured, and typically is provided as external metadata. In plasma physics, experimental data for the poloidal flux within toroidal plasma devices is typically mapped to a flux-based non-orthogonal curvilinear coordinate system. The resulting coordinate system is then used to evaluate the stability of the configuration to magneto-hydrodynamic instabilities. The generation of the flux-based coordinate system requires the application of an algorithm. The data quality of the resulting interpretation of the plasma stability is strongly tied to the accuracy of the toroidal geometry representation. In this case a procedure that is applied by a user determines the data quality.

Scientific data sets are generated with well-defined **physical variables**. The set of variables desired by a user may require combinations of the variables present within the data set. An example of a system that extracts data from a data set based on physical variables is the OpenDAP and THREDDS environment. It is possible to extract physical variables from a data set, without retrieving the entire file. The quality of the physical variables depends more on the transformations that may be needed to convert to desired quantities. Thus the conversion from velocity to vorticity depends on how well the conversion routine approximates the curl operation, which in turn depends upon the spatial resolution provided by the coordinate system and the number of spatial points needed to implement a curl operation. The transformation function accuracy is even more important when interpolating data for use in differential equations. In practice, data analyses can introduce numerical artifacts if the degree of the interpolation function is not commensurate with the solution order of the differential equation. In this case, data quality requires self-consistent treatment of both data and analyses by the user of the data collection.

The quality of the **physical units** depends mainly on the consistency across the data collection. If some variables are in feet/pound/second units and some are in meter/kilogram/second units, the data will easily be misinterpreted and may lead to incorrect analyses. An example of poor physical units quality was the crash of a Mars rover.

The quality of the **measurement accuracy** (number of significant digits) is important for determining the allowed transformations. The data accuracy may be so poor that the desired physical effect cannot be separated from noise in the data. However, averages of the data may be sufficient to track changes over long time periods, or to track effects that appear from superposition of many data sets. An example is the association of quasars with galactic centers, by superimposing thousands of quasar images. In this case, the users of a data collection could generate meaningful research results even though each individual data set lacked the required measurement accuracy.

The **provenance** of the data needs to include descriptions of all processing steps that were applied to the data. The standard example is the processing of satellite data by NASA. The data have to be calibrated, turned into physical variables from raw sensor data, and then projected onto a coordinate system. Each processing step requires the application of a procedure that significantly transforms the data. Assertions about data quality are then driven by the accuracy of the transformations, as well as by the original accuracy and resolution of the raw data. In this case, the quality assessments are done by the creators of the data collection. However, if a calibration is revised, the data quality becomes highly suspect and the transformations must be re-done for new assertions about data quality.

Transformations applied to data also may depend upon **physical approximations** that are used to simplify the analysis. The physical approximations may be associated with type of physical flow (compressible or incompressible), type of equation of state, type of assumed particle distribution functions, etc. For a consistently derived data set, similar physical approximations need to be applied across all transformations performed upon the data.

A related issue is the set of physical constraints that are enforced when the data are manipulated. Do the numerical algorithms enforce physically conserved properties, such as energy, mass, and momentum? A simple example is the projection of telescope images to a standard coordinate system. To conserve the intensity, spherical trigonometric functions need to be used to project each pixel. This was done in the 2-micron All Sky Survey to generate a unifying mosaic of the night sky. If the algorithms had applied trigonometric functions, the intensity would have been blurred.

Another set of physical constraints is the set of assumptions for how missing data points will be handled. Is the missing data marked as missing, or are interpolation functions used to approximate the missing data? A standard example is the generation of a uniform world weather model that incorporates observational data. A numerical weather simulation is run forward in time based on the observations for 6 hours. The result is then compared with new observations. Forcing functions are derived such that running the simulation a second time will generate the actual observations seen at the end of the 6-hour run. This interpolates the weather onto a uniform grid in space and time, effectively interpolating across all missing data points. The interpolation accuracy depends upon the physical approximations that were made in the weather model. Each time the physical approximations are improved, a re-analysis is needed to generate a better interpolation onto the uniform grid in space and time. These analyses are typically performed by the creators of the data collection.

A second example is the analysis of radar data to generate precipitation estimates. Re-analyses are done based on improvements in the physical model for reflection of radar waves by water. The quality of the data set is driven by the quality of the physical model.

Semantics (and ontologies that describe how semantic terms are related) can lead to data quality control issues. Each domain defines a standard set of semantic terms that describe physical phenomena. Each research group tries to refine that description of domain knowledge to improve the understanding of the underlying physical world. The semantics used by a research group evolve to track their improved understanding of physical reality. Thus semantic terms, as used by a research group, may have nuances of meaning that are not known to the rest of the community. This results in different definitions of data quality, based on the understanding of the underlying physics.

5. SUMMARY

Data quality curation inherently requires the application of procedures to verify or create required data set properties. An analysis of the data quality of a collection requires a detailed understanding of the curation procedures. In policy-based data management systems, the data curation procedures can be preserved, and re-executed in the future to verify an assertion about the data quality that is made by the creators of the collection. However, the users of a data collection may have different required properties for data quality that in turn depend upon application of additional procedures. An assessment of data quality by the users of a collection may generate a different interpretation of the relevance of the data for their research project. Data quality assessments require a mapping between assertions made by the creators of a collection, and the collection properties needed by the users of a collection. This requires the ability to control application of procedures, sharing of procedures, re-execution of procedures, and preservation of procedures. Data quality curation can be mapped to the procedures that are used to verify assertions about a data collection.

6. ACKNOWLEDGMENTS

The topics presented here were supported by the NSF Office of Cyberinfrastructure OCI-0848296 grant, “NARA Transcontinental Persistent Archive Prototype”, (2008-2012), the NSF ITR 0427196 grant, “Constraint-based Knowledge Systems for Grids, Digital Libraries, and Persistent Archives” (2004-2007), the NSF SDCI 0721400 grant, "SDCI Data Improvement: Data Grids for Community Driven Applications” (2007-2010), the NSF OCI-1032732 grant, "SDCI Data Improvement: Improvement and Sustainability of iRODS Data Grid Software for Multi-Disciplinary Community Driven Application," (2010-2013), and the NSF Cooperative Agreement OCI-094084, “DataNet Federation Consortium”, (2011-2013).

7. REFERENCES

[1] Rajasekar, R., Wan, M., Moore, R., Schroeder, W., Chen, S.-Y., Gilbert, L., Hou, C.-Y., Lee, C., Marciano, R., Tooby, P., de Torcy, A., and Zhu, B.. 2010. *iRODS Primer: Integrated Rule-Oriented Data System*, Morgan & Claypool. DOI= 10.2200/S00233ED1V01Y200912ICR012.

[2] Ward, J., Wan, M., Schroeder, W., Rajasekar, A., de Torcy, A., Russell, T., Xu, H., Moore, R. 2011. *The integrated Rule-Oriented Data System (iRODS 3.0) Micro-service Workbook*, DICE Foundation, November 2011, ISBN: 9781466469129, Amazon.com