The International Journal of Digital Curation
Volume 8, Issue 2 | 2013

Evolving Persistent Archives and Digital Library Systems: Integrating iRods, Cheshire3 and Multivalent

Reagan Moore and Arcot Rajasekar,
University of North Carolina

Paul Watry, Fabio Corubolo, John Harrison and Jerome Fuselier,
University of Liverpool

Abstract

This paper describes work undertaken by Data Intensive Cyber Environments Center (DICE) at the University of North Carolina at Chapel Hill and the University of Liverpool on the development of an integrated preservation environment, which has been presented at the National Coordination Office for Networking and Information Technology Research and Development (NITRD), at the National Science Foundation, and at the European Commission. The underlying technology is based on the integrated Rule-Oriented Data System (iRODS), which implements a policy-based approach to distributed data management (Rajasekar et al., 2006). By differentiating between different phases of the data life cycle based upon the evolution of data management policies, the infrastructure can be tuned to support data publication, data sharing, data analysis and data preservation. It is possible to build generic data management infrastructure that can evolve to meet the management requirements of each user community, federal agency and academic research project. In order to manage the properties of the data collections, we have developed and integrated scalable digital library services that support the discovery of, and access to, material organized as a collection.

The integrated preservation environment prototype implements specific technologies that are capable of managing a wide range of preservation requirements, from parsing of legacy document formats, to enforcement of preservation policies, to validation of trustworthiness assessment criteria. Each capability has been demonstrated and is instantiated in multiple instances, both in the United States as part of the DataNet Federation Consortium (DFC) and through multiple European projects, primarily the FP7 SHAMAN project.

International Journal of Digital Curation (2013), *8*(2), 47–67. http://dx.doi.org/10.2218/ijdc.v8i2.268

Introduction

A primary goal of the preservation community is to provide real software that supports production systems managing petabytes of data and hundreds of millions of files. The strategy is to organize the distributed data into shared collections and then manage the properties of the shared collections. The properties depend on the type of data management application but can include integrity assertions, authenticity assertions, chain of custody assertions, trustworthiness assertions and scalability mechanisms.

An increased level of attention is needed on how to make the system work at scale when data records are distributed across institutions, administration domains and continents. The approach taken is to focus not on an individual record, but to focus on the collection which provides a context for the records. What assertions can we make that will hold true for all records in the archival collections? How do we make and enforce uniform properties across all records in the archival collections?

We recognize immediately that for distributed data management, we need to ensure that the data grid manages the properties needed to make assertions about the archival collection. We cannot rely on the remote storage locations, since they use different protocols, do not provide the support needed for checksums or replication, do not provide mechanisms to enforce management policies, and do not support descriptive metadata.

In order to demonstrate the viability of the approach, we require a demonstration of how data grids (ostensibly created to support organization of shared collections) can be used to support digital library services, preservation environments and data processing pipelines, as described in this paper. This requires the development of generic infrastructure (a ‘virtual data grid’) that can support all types of data management applications.

We realize that this approach requires the mechanisms needed to manage technology evolution for preservation environments. At the point in time when new technology becomes available, both the old and new technologies are present. An integrated preservation environment is one that supports interoperability between the different versions and enables the migration of records from the old technology to the new technology. As part of this development, we can demonstrate how digital library services (access and discovery) can be encoded as procedures that are applied to collections under the control of management policies. This results in the realization that workflows can be distributed between the storage systems, computer servers and the display engines.

Through use of virtual data grid technology by several hundred projects around the world, we came to realize that the major difference between the projects were the set of management policies that were used to enforce collection properties. This has led to the concept of a virtual data grid technology that could support management policies as computer actionable rules, while assembling procedures from sets of micro-services that could be executed at remote storage locations.

The result was a generic infrastructure model, iRODS, that can support all data management applications, including preservation. We note that the pace of technology evolution is sufficiently rapid that all data management initiatives need to be concerned about preservation. Preservation is defined traditionally as the enforcement of authenticity, integrity, chain of custody and original arrangement on the archived records. Policy-based systems implement procedures that enforce each preservation property. The procedures interface between the desired preservation properties and the protocols used to interact with the changing external storage technologies. We therefore define the concept of preservation as the management of technology evolution while communicating with the future.

We describe the application of the iRODS virtual data grid system in terms of an engineering activity, with the expectation of building robust, reliable, distributed data management software. All of the technologies we describe have appeared in some context before. However, no one else has attempted to integrate them to achieve the objective of assembling a shared collection from distributed data. The DICE Foundation has written the software code that provides the framework. Into that framework, as part of the SHAMAN integrated project, the University of Liverpool has plugged in digital library and knowledge management technologies, including the Multivalent Browser technology and the Cheshire3 information retrieval system, each described below.

The integrated preservation environment has generated a highly extensible data management environment. When we add new rules and procedures, we can simultaneously add the new state information required to track the result of applying the rules. This means the iRODS system is capable of internal evolution. We can manage a sub-collection using the old rules, procedures and state information, and migrate the records in the sub-collection to a new sub-collection governed by a new set of rules, procedures and state information.

iRODS Architecture Overview

The transformation of preservation policies into computer actionable rules is the essential capability that is needed to manage data collections that will aggregate hundreds of petabytes of data. The capability to execute rules is provided by the generic iRODS[1] system, which makes it possible to automate administrative functions (such as distribution, retention, disposition, replication and synchronization), enforce management policies (such as the formation of AIPs, the generation of audit trails and the creation of error reports), and validate assessment criteria (such as integrity, authenticity, chain of custody, original arrangement and trustworthiness). The policies that are used for preservation form a continuum with earlier stages of the data life cycle. The mechanisms to manipulate the data remain the same through the data life cycle, but the management policies evolve as the user community broadens. The policies required by the original research project are broadened to meet data publication expectations by the discipline, and then broadened to meet preservation requirements for use by future generations. Preservation policies can be interpreted as the most stringent data management requirements, because they need to ensure that

[1] iRODS: http://www.irods.org

the required context (representation information) is available for use by a future, undefined community.

The iRODS data grid is software infrastructure that is installed at each storage location. The iRODS server consists of the software that translates from the iRODS operations to the access protocol required by the specific storage system, a distributed rule engine and a distributed rule base. The types of storage systems that can be used include Unix file systems, Windows file systems, High Performance Storage System (HPSS) tape archives, Sam-QFS tape archives, cloud storage systems and object storage systems. One of the iRODS servers also includes a centralized metadata catalogue through which all state information is deposited in a relational database.

At each storage location, iRODS uses a distributed rule base to control the procedures that manipulate the records. The rules are cast as event/condition/action-chains/recovery-chains. The procedures (action-chains) are composed from micro-services that encapsulate specific data manipulation functions. At present, 317 micro-services are provided for composing the procedures needed to implement a desired policy in iRODS version 3.3.

Research on the policies governing preservation has been conducted for the last 15 years years under funding provided by the National Archives and Records Administration through the National Science Foundation and the European Commission. The fundamental preservation requirements can be viewed as a sequence of four context transitions:

1. Extraction from the creation environment,
2. Export into future preservation environments,
3. Validation of preservation assessment criteria based on policies applied in the past,
4. Migration onto new management policies.

During each transition, the intent is to preserve preservation attributes for authenticity, integrity, chain of custody, original arrangement and trustworthiness.

The iRODS data grid is designed to work at the scale of hundreds of millions of files and petabytes of data (Rajasekar et al., 2003). For some federal agencies, a centralized metadata catalogue is required for maintaining control of deposited material. To improve scalability, master-slave metadata catalogues can be created, with all writes to the master metadata catalogue. Read accesses on the data grid can be done from a slave metadata catalogue. A second approach for improving scalability is to use replication capabilities of the underlying choice of database technology. The metadata catalog can be implemented in a Postgres, Oracle or MySQL database. The pgpool-II mechanism can be used to distribute metadata across multiple Postgres database instances. Finally, through federation of independent data grids, each with their own metadata catalogue, preservation environments can be extended to arbitrarily large archives. This is the approach taken in the National Archives (NARA) TPAP project, which federates seven independent data grids.

SHAMAN Integrated Digital Library Technologies

The EU SHAMAN[2] project (Sustaining Heritage Access through Multivalent Archiving) used the iRODS virtual data grid as a tool for developing next generation digital library services that support access, presentation, and discovery or analysis of archival collections independently of the underlying storage infrastructures. The conceptual model focused on the creation of an end-to-end system based on workflows that could demonstrate the integration of digital library and persistent archive services, including support for documents, media, data, services, provenance and state. The overall conceptual scheme is set out by the NARA Persistent Archive prototype, which describes 'characterizing preservation processes', including not only the bits, but also logical structures and relationships.

The requirements of this project in many ways reflect the aims and objectives of the NARA Persistent Archive prototype, but with an increased emphasis on data discovery and persistent parsing capabilities. The project recognized these as additional elements required to make an archive more usable.

The SHAMAN project was based on the integration and orchestration of three different technologies to highly scaled and distributed data. These technologies are the the iRODS data grid to manage the storage workflows, the Cheshire3[3] digital library system to manage the processing and analysis of the archive, and the Multivalent[4] browser to manage the presentation of the archived collections.

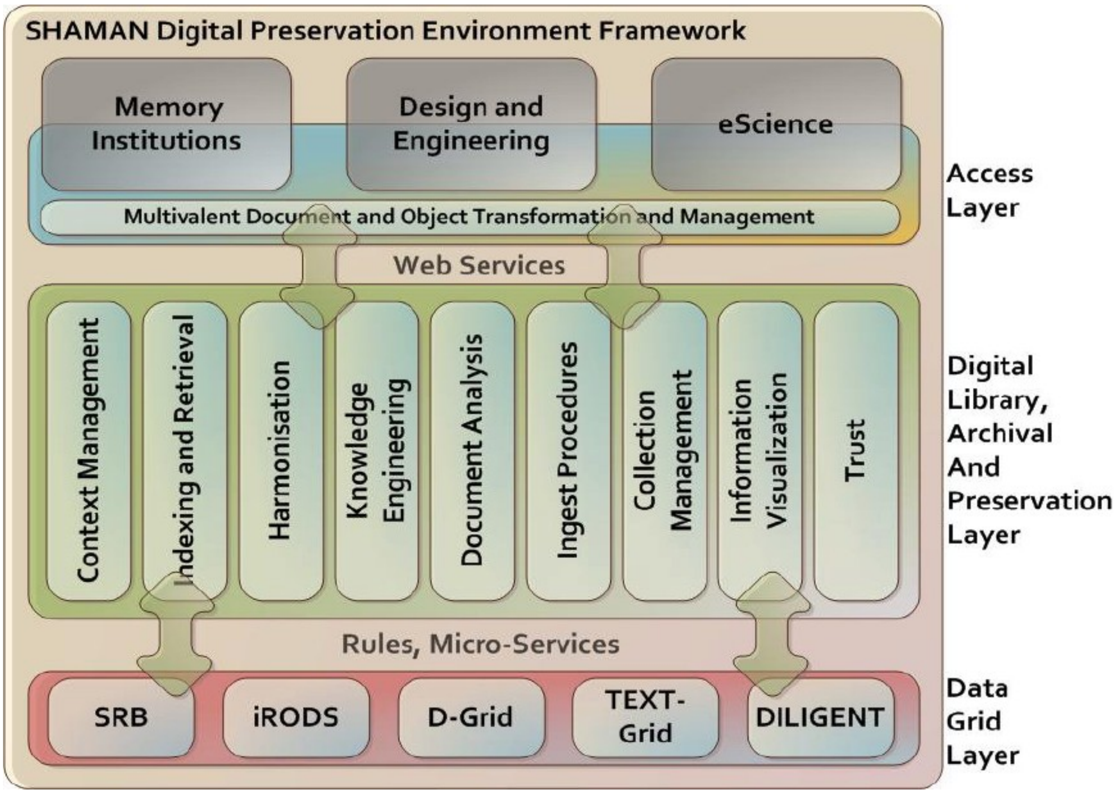

Figure 1. The SHAMAN Digital Preservation Framework.

[2] SHAMAN project: http://www.shaman-ip.eu

[3] Cheshire3: http://www.cheshire3.org/

[4] Multivalent: http://multivalent.sourceforge.net

The project methodology recognizes a series of ordered steps required for preservation; in effect, these are characterized as an encapsulation of the iRODS processes through the addition of digital library services ported onto the storage. The steps consist of pre-ingest, wherein the data and metadata to be preserved are assembled; ingest into the preservation system, in which the searchability data is extracted; discovery, incorporating search interfaces (both human and computer); and presentation and annotation technologies, based on the use of the Multivalent object model, described below.

- **Step 1: Pre-ingest** – This consists of the assembly and description of the objects to be preserved, resulting in the creation of a Submission Information Package (SIP), which contains URL references to the data and all and any submitter-created metadata. The data itself must be available to the archive service by resolution of the the URL references in the SIP. This may be achieved either by exposing it via HTTP or FTP, or by uploading it to a pre-ingest staging area within the iRODS system. The implementation of the SIP is based on the Open Archives Initiative Object Reuse and Exchange (OAI-ORE) Model. This is a widely used digital library standard, supporting the Resource Description Framework (RDF) specification metadata model.
- **Step 2: Ingest** – This consists of an archive-ready submission information package (SIP) that incorporates information relating to policy-based management of the archive, including authentication, authorization, validation and storage. The strategy is to adopt SWORD[5] as the standard protocol for depositing a SIP into the archive. The ingest phase is then used to retrieve the submission payload via network location (FTP, HTTP) references in the SIP, and generate preservation and discovery metadata, which provides the basis for discovery and analysis processes. This ‘deposit-by-reference’ approach enables the ingest of multiple, arbitrarily large files without exceeding the limitations of the submission protocol. The process generates an Archive Information Package (AIP).
- **Step 3: Discovery** – This is based on use of the Cheshire3 digital library system to support processing workflows which are integrated with the storage side workflows of the iRODS system. The Cheshire3 system is designed to demonstrate the effectiveness of digital library technologies to support large-scale distributed storage requirements. The Cheshire3 indexes and software are both archived as iRODS collections that can be called up by the system to automate the processes required to support resource discovery across domains and formats.
- **Step 4: Presentation** – This is based on the deployment of the Multivalent object model, described below, that satisfies the ability to present and manipulate data with few infrastructure dependencies. As part of the project, we engineered specific annotation capabilities, which supported shared, distributed annotations that are semantically anchored, enabling them to be applied across formats.

[5] Simple Web Service Offering Repository Deposit (SWORD): http://swordapp.org

In this scenario, maintaining the ability to interpret the data as meaningful structures and relationships, and displaying accurate visual representations of them over time, represents a primary contribution of the project based on the use of the Multivalent technology, and the implemented Fab4 Browser[6]. The approach reflects conceptual advances emerging from the ongoing developments of the Data Format Description Language (DFDL) and Defuddle – a generic parser that can use DFDL-style format descriptions to extract logical structures from ASCII or binary files written in those formats. The Multivalent object model can be understood as an interpretor of these descriptions (i.e. as a means of characterizing the structure of binary and character encoded files and data streams) so that their format and structure can be exposed.

Throughout the project, the Multivalent object model was used to characterize heterogeneous data resources in the iRODS virtual grid environment, including media, document and CAD format files. We specifically did not aim to create a generic data representation tool; rather Multivalent was used to present existing formats in an actionable manner that made the data useable in their current format. Access to legacy data formats can be demonstrated through the Multivalent browser technology. This is software that emulates the public data parsing and manipulation capabilities in a transportable language (Java). By managing a parser for each record format type, it is possible to ensure the ability to manipulate and display records in the future. When the record is accessed, the legacy format is parsed using the Multivalent browser technology (media engine) and transformed into the required display format. This avoids the need to apply transformative migrations to entire archives that can introduce data loss and overheads, and simplifies long-term administration.The parsed file can then be accessed, queried and integrated, regardless of its data format. We can also use the Multivalent tool to describe the format of a data set as a XML schema, for example the Data Format Description Language (DFDL) schema[7]. The SHAMAN integrated project focused on the use of the Multivalent object model and Fab4 browser as a mechanism to apply future display and manipulation mechanisms to data that comes from the past (Phelps and Watry, 2005).

A persistent archive needs both the management of the preservation environment and a stable characterization of the data format (provided by Multivalent). The preservation environment provides procedures that can be transported into the future onto new technology while preserving the ability to parse and manipulate the records using the Multivalent tool.

An essential capability explored within the SHAMAN project is the concept of on-demand transformative migration of records. For collections that are petabytes in size, it may not be feasible to migrate records to new data formats each time a new standard is published. Instead, SHAMAN pursued an approach based on persistent objects:

- The original records format is preserved without any changes;
- An infrastructure independent method is provided to parse the data format using the Multivalent Browser technology;

[6] Fab4 Browser: https://code.google.com/p/fab4browser/

[7] Data Format Description Language schema: http://www.ogf.org/dfdl

- Parsers are written that migrate a record from an original format to the new desired display format;
- When an object is accessed, the transformative migration to the new display format is done on-the-fly.

The SHAMAN project also demonstrated the concept of multiple independent policies/procedures/workflows for managing preservation. Three different levels of processing can be applied to records:

1. External workflows that govern the record accession process – an example is the Producer Archive Workflow Network (PAWN), which is used to manage the chain of custody as data are ingested;
2. iRODS server-side workflows that govern administrative functions and validation of assessment criteria;
3. Client-side display workflows that manage the transformations needed to display and manipulate records – these can be implemented in the Cheshire3 analysis system on records extracted from the iRODS data grid.

The development of these digital library services, using the iRODS virtual data grid, created an end-to-end data curation environment and resulted in two major outcomes:

1. The engineering of real software used in production by national and international communities;
2. A demonstration of how the many different types of data management applications could be supported by generic common infrastructure.

The work has promoted a greater understanding of the fundamental concepts required to build a viable preservation environment when the component technologies are evolving. Thus, we need to specify the essential difference between data, information and knowledge management, and how to build infrastructure that supports data (bits), information (labels on structures imposed on bits), and knowledge (relationships between labels). This has led to further use of the virtualized data grid and digital library technologies as part of the DataNet Federation Consortium (DFC) and multiple European framework projects. Related projects are based at the Odum Institute for Research in Social Science[8], the Renaissance Computing Institute (RENCI)[9], the Australian Research Collaboration Service (ARCS)[10], and EnginFrame[11].

[8] Odum Institute for Research in Social Science: http://www.irss.unc.edu/odum

[9] RENCI: http://www.renci.org/

[10] ARCS: http://le.unimelb.edu.au/research/arcs.html

[11] EnginFrame: http://www.nice-software.com/products/enginframe

Applications

An intent of our research is to illustrate the wide range of data management applications that can be supported from generic, policy-based data grid infrastructure. By varying the set of policies and procedures (expressed through computer-actionable rules), all phases of the data life cycle can be controlled. The end goal is a reference collection that can be used to document events, or serve as a resource against which future research can be compared, or serve as a knowledge base for predicting the result of future economic decisions. For the reference collection to be trusted, the preservation environment needs to document the policies under which the reference collection was managed, the procedures that were applied and the assessment criteria that were used to validate compliance. The iRODS data grid provides an explicit characterization of polices as computer-actionable rules, of procedures as computer-executable micro-services, and of assessment criteria as queries on state information and parsing of audit trails. It is now possible to build a data management system that manages all phases of the data life cycle.

Infrastructure Independence

The extraction of records from the creation environment and their import into the preservation environment is an example of infrastructure independence (Moore, 2006). The archivist controls the properties of the record within the preservation environment, including the names assigned to the records, the names assigned to archivists, the names assigned to storage resources, and the policies and procedures that govern the management of the records. The concept of infrastructure independence is important when building a preservation environment that distributes records across multiple types of storage systems and across the federation of multiple data grids. Within a data grid, the policies govern the management of records residing in multiple administration domains. The policies can be enforced on top of the local administration policies. Examples include replication of data across multiple storage systems, association of descriptive and provenance metadata with each record, and uniform creation of AIPs for all records independently of where they are stored.

Additional policies govern the federated environment. These policies control the replication of data between the independent data grids, the synchronization of records between a deep archive and an access environment, and the identification of the authentic source. What is implemented is a control hierarchy, with each additional level governed by rules that enforce more systemic policies.

- **Preservation environment policy** corresponds to federation policies across data grids. This is the highest policy level.
- **Archive policy** corresponds to policy governing an archives. This governs how a record series will be managed.
- **Resource policy** corresponds to policies specific to a storage device or administrative domain. These govern how the specific storage device is managed.

By changing the preservation environment policies, a system can be constructed that promotes data sharing between institutions, or that provides sustainability mechanisms for ensuring continuity of the preservation environment as funding transitions from an original support institution to a new support institution.

The iRODS data grid framework is instrumented to automatically invoke policies associated with specific data manipulation operations. Examples are the automatic invocation of a policy before ingestion of a file (typically for additional authorization checks), or automatic invocation after ingestion of a file (typically to generate required derived data products). Policies are invoked at 71 locations within the framework, including put, get, move, copy, replication and registration of files; creation, modification, deletion of users; creation, modification and deletion of storage resources; creation, modification and deletion of collections; and creation, modification and deletion of metadata. Policies also control the degree of parallelism used in data transfer, use of external identity management systems, automated indexing of the ICAT metadata catalogue, automated purging of files, etc.

The iRODS data grid implements data virtualization (management of collection properties independently of the storage system), trust virtualization (management of authentication and authorization independently of the storage system), and policy virtualization (enforcement of management policies independently of the storage system). Each type of virtualization requires that iRODS manage the names of the corresponding entities (users, files, storage systems, rules, micro-services and state information). The iRODS data grid maps from the physical name for the entity to a logical name that is used as a persistent, global identity. Each storage resource has a logical name as well as a physical IP address. The storage logical names can be organized into storage resource groups. Collective operations can then be assigned to the storage resource group, such as load levelling or fault tolerant data ingestion. Similarly, user names can be organized in groups and files can be organized in collections.

Security is imposed as a constraint between any two logical name spaces. Thus traditional access controls are implemented as a constraint between the user name space and the file name space. Pinning of data to a storage system is implemented as a constraint between the file name space and the storage name space. Quotas can be implemented as a constraint between the user name space and the storage name space. Part of trust virtualization is support for multiple types of authentication systems, including a challenge-response mechanism, Grid Security Infrastructure public key certificates, Kerberos-based authentication and Shibboleth. The goal is to manage authentication and authorization across multiple types of authentication environments.

Data Life Cycle

There is a driving purpose behind the formation of a shared collection, and an intended user community. The purpose defines the properties that the collection should possess. Examples include authoritativeness (coming from a recognized source), authenticity (coming with a provenance chain), completeness (comprising all of the required data for the specific project), consistency (uniform data format and set of descriptive metadata), and usability (manipulated by a standard set of tools). The community that forms the collection builds a social consensus on the set of properties that a user can expect from the collection.

The implementation of the collection requires the creation of policies to enforce the desired properties. The policies usually control ingestion of records into the collection to ensure that the required information for each record is provided. However, policies can also control access, redaction and administrative functions, such as integrity checks. The social consensus also needs to define the required state information that will be managed by the system. The iRODS data grid (version 3.3) manages 338 state information attributes about users, files, collections, resources, rules and micro-services. Additional state information attributes can be added that are specific to new policies.

A new stage of the data life cycle is entered when either the user community broadens (and thus requires a new social consensus) or the driving purpose changes. The evolution of the collection can then be tracked through the new policies and procedures that are required to enforce the new set of properties. In practice, preservation in a reference collection is usually the most severely controlled environment, as the collection must be usable by a future generation. This means that no assumptions can be made about collection-specific knowledge. All information that is needed to interpret and use the collection (including parsing routines) must be explicitly preserved.

In our prototype, each phase of the data life cycle is illustrated using an existing collaboration. The iRODS data grid is used to support collection creation, data analysis pipelines, data publication and data preservation, using the digital library services provided through SHAMAN.

Collection Creation

In practice, most policies are related to the control of the ingestion of records into the shared collection. Policies can specify what is needed for each file and how records can be bulk loaded through use of an aggregation mechanism. Files can be aggregated in containers, such as tar files, to simplify data movement. Metadata can be aggregated in XML files to simplify bulk loading.

An example is the Odum Institute’s preservation of social science data. The data comprise answers to questionnaires from surveys. They can be extracted from the Odum collection, converted to standard XML, and reduced to the subset appropriate for discovery-based browsing. This process was automated within iRODS through the construction of micro-services that apply XSLT transformations to XML files and that bulk load XML files into the iCAT catalogue. For the Odum collection, an XML file containing all of the Odum metadata was archived, and an identified subset was made accessible through a query interface to support browsing.

Some of the challenges included maintaining privacy concerns on the metadata, managing access rights, and supporting federation with other repositories to implement federation-based long-term sustainability. Odum has sustained each collection by identifying multiple communities that require access, and finding new communities whenever an original community can no longer provide support.

Through iRODS federation of independent data grids, a similar approach to sustainability is possible, with each institution managing a separate instance of the collection, but through the federation ensuring long-term support across multiple institutions.

Processing Pipelines

An optimal workflow separates processes into those with low complexity (small number of operations compared to the size of the file in bytes) and those with high complexity (large number of operations compared to the size of the file in bytes). Low complexity operations can be performed at the storage location through server-side workflows, such as iRODS. High complexity operations are performed at a compute engine through client-side workflows, such as Kepler[12] or Taverna[13]. We give two examples of processing workflows.

The NSF MotifNetwork project[14] organizes proteins into their constituent domains. This requires the ingestion of information from multiple existing databases, the transformation of the data into a form suitable for analysis, the execution of multiple processing steps, and the organization of the output files for re-use in future computations. The analyses can generate thousands of files.

Jeffrey Tilson developed a workflow interface between iRODS and the Taverna workflow system[15]. Taverna processes were created to get a file, put a file, make a directory, change directory, list files, and replicate files. This made it possible to automate MotifNetwork data analyses and improve research turnaround time by a factor of ten. Automation of the processing pipeline greatly decreased the effort required to do the research.

In version 3.3 of the iRODS software, support is provided for registering workflows as a data object. Clicking on the registered workflow causes the execution of the workflow, the automated tracking of workflow provenance, and the automated versioning and archiving of workflow results. The workflow, the input files and the output files can be shared. It is possible for a collaborator to change an input file, re-execute a workflow and compare output results. The management of workflows and workflow provenance make it possible to support reproducible data-driven research.

The implication for preservation is that the processes that manage ingestion or validation of assessment criteria can be registered as workflows that are automatically tracked by the preservation environment. In particular, the results of each assessment validation workflow can be versioned and saved. An archivist now has the mechanisms to validate communication from the past. Assertions made by prior archivists about processes applied to the preservation environment can be evaluated and re-executed if needed. This closes the loop on the interpretation of preservation as communication with the future. An archivist in the future can manage and validate assertions made by archivists in the past.

[12] Kepler: http://kepler-project.org/

[13] Taverna: http://taverna.sourceforge.net

[14] NSF MotifNetwork: http://www.renci.org/focus-areas/biosciences-health/motifnetwork

[15] See: http://www.tacc.utexas.edu/

Data Grids

A data grid can manage enormous amounts of data (petabytes in size) that are distributed across multiple types of storage systems. In common practice, a data grid administrator manages the data grid and performs such tasks as adding a new resource, verifying replication of records, validating integrity, synchronizing local and data grid resources, adding users, and tracking down problems (resource off-line, network down, corrupted data). The management effort is onerous at the petabyte level in a distributed environment, because there is always a problem somewhere.

Policy-based data management systems provide the asynchronous support mechanisms needed to automate many administrative tasks. These tools consist of rules that are periodically executed to verify assessment criteria and repair problems that are discovered. A simple example is the periodic synchronization of two federated data grids to ensure that the second data grid holds a true copy of the contents of the first data grid. Any files that are not synchronized will be automatically corrected by the next iteration of the periodic rule.

A major objective is to identify the policies that should be enforced. Each community has specific criteria that they must satisfy. For example, the NSF Science of Learning Center at the UCSD Temporal Dynamics of Learning Center enforces Institutional Research Board (IRB) approval policies. A data grid was created that linked storage resources at UCSD, Brown University, Rutgers, and Vanderbilt. Records were registered into the Temporal Dynamics of Learning Center data grid[16] to enable collaborative research through sharing of data. For records that contained human subject data, the institution's IRB policy on data distribution had to be enforced.

At UCSD, an administrative database was created that recorded all of the IRB approval decisions. A micro-service was written that could read the database and extract the distribution and access controls. A rule could then periodically execute the micro-service and set the appropriate distribution and access controls on the files within the iRODS data grid. Since the files were initially registered into the data grid with only the owner of the file given access, this ensured that all access by other researchers had been appropriately reviewed and granted.

In practice, the policies for building collections, publishing data, preserving data, and analyzing data are all different. They control different processing steps, such as metadata extraction, metadata registration, or creation of derived data products. Since iRODS supports multiple versions of a rule, it is possible to define a separate policy for a user group, data collection or file type by modifying the condition within the rule. The iRODS data grid will check the multiple versions of a rule to find the first one that applies, and will then execute the action-chain. Each rule version can invoke the set of micro-services that are appropriate for that community.

We also observe that each community typically prefers to use a specific access client. This forced the development of the ability to map from the protocol and tasks requested by a client to execution of generic iRODS rules. The rules invoke micro-services that execute standard I/O operations that are based on Posix functions.

[16] Temporal Dynamics of Learning Center data grid: http://tdlc.ucsd.edu/

The Posix functions are mapped to the protocol required by a storage system by an iRODS storage resource driver. This ensures that any client can be ported onto the iRODS system, without having to modify either the micro-services or the storage resource drivers. The variety of clients that access the iRODS data grid is driven by the wide variety of user communities. They include (along with the requesting community) web browsers, WebDav (ARCS), FUSE user-level file system (Teragrid), Taverna (MotifNetwork) and Kepler workflow systems, Fedora[17] digital library, DSpace[18] digital library, Windows browser (NARA), Python load library, C I/O redirection library (NASA), JARGON Java I/O class library, etc.

A simple example of a policy from the NASA Center for Computational Science is the automated replication of a file on input (NASA, 2009). A rule is defined that automatically creates the replica and stores the associated state information on each file ingestion. The policies can retrieve information from the iCAT catalogue to decide what action to perform. Rules can be executed remotely at multiple storage locations, and rules can be created that invoke web services.

A second example is the replication of an astronomy collection between the RENCI and the TACC data grids. This used the federation support mechanisms within iRODS. Two administrative commands were issued at each data grid; one command to establish the existence of the other data grid, and a second command to set up an account in the remote data grid for a user from the first data grid. Once trust was established between the data grids, a user from the RENCI data grid could store a file in the TACC data grid. The RENCI data grid user was identified as a foreign user by TACC, and rules governing the allowed operations by the foreign user could then further restrict the set of operations that could be performed. The astronomy image collection that was replicated was the Digital Palomar Observatory Sky Survey[19], consisting of three terabytes of data.

The type of federation is controlled by explicit policies. Production examples range from chained data grids in which data is pulled successively to each data grid in the chain (NOAO[20]); to master-slave data grids in which data is only written to the master data grid, but replicated to each slave data grid for reading (NIH); to a central archive to which each grid replicates records (UK e-Science data grid[21]); to a deep archive that pulls data into a preservation environment (NARA); to the above TACC replication data grid.

A final example is the integration of cloud storage resources into the iRODS data grid. This required the development of a driver for the Amazon S3 interface[22], and the construction of a compound resource for interacting with the cloud. On storage of a file, a copy is first created on a disk cache. The iRODS rules are applied, and the file is replicated into the cloud storage. This approach was necessary to handle the parallel I/O streams that iRODS normally uses for large files, to ensure that partial I/O can be done on the file, and to support the client-side workflows. Normally, a cloud storage

[17] Fedora: http://www.fedora-commons.org/

[18] Dspace: http://www.dspace.org/

[19] Palomar Observatory Sky Survey: http://www.astro.caltech.edu/~george/dposs/

[20] NOAO: http://www.noao.edu/

[21] UK e-Science data grid: http://www.escience-grid.org.uk/

[22] Amazon S3: https://s3.amazonaws.com

resource only allows file put and file get, without application of a server-side workflow.

Digital Library

Multiple communities are building digital libraries on top of the iRODS data grid through integration with the Fedora digital library middleware. They use the Fedora object model to characterize the required provenance and descriptive metadata, and to establish relationships between records. They establish a web-accessible portal on top of Fedora to control presentation of the information, and support search and browsing functions.

A good example is the EPA Community Modeling and Analysis System (CMAS)[23]. EPA air quality data sets were loaded into an iRODS data grid for access through their portal. A user could search for desired data sets by keyword, by year, or by model/resolution/file type. The user could display metadata, and download both metadata and data.

Digital libraries usually have to manage large numbers of small files. An example was the National Science Digital Library[24] that archived web crawls of educational material. To efficiently handle the small data sets, iRODS provided mechanisms to aggregate records into a tar file, and then store the tar file. Information about each record was retained by iRODS, enabling the extraction of the desired record from the tar file. The interface used by iRODS to manage this was a Structured Information Resource Driver. This interface maps from iRODS operations to the protocol required to manipulate the structured information (the tar file in this case). Additional Structured Information Resource Drivers have been written to interact with directories (equivalent to mounting a remote directory into the iRODS data grid). The implication is that some of the information that is needed to manipulate the record is stored within the structure information. iRODS queries the structured information resource to find out the information needed for subsequent operations. This mechanism promises to become a standard approach for interoperability between different types of data management systems at the file manipulation level.

Scientific digital libraries can also be constructed on top of iRODS. In many cases, the operations required by the scientific digital libraries are similar to those provided by a distributed operating system: remote job execution, remote information exchange, remote job scheduling, remote data management. The iRODS data grid implements servers that perform these functions, from queuing of rules for execution, to high-performance message passing for exchanging information between micro-services. A specific example is the RENCI data grid integration of the Big Board interactive visualization system with iRODS. Ortho-photos can be pulled from the iRODS data grid for visualization on the Big Board display environment.

[23] CMAS: http://www.cmascenter.org/

[24] National Science Digital Library: http://nsdl.org/

Preservation Environment

The NARA TPAP project requirements drove much of the iRODS development. The requirements are based on four preservation perspectives: extraction of records from their creation environment, migration of the records onto new technologies to enable communication with the future, specification of current preservation policies and procedures to enable future archivists to assess compliance, and validation of policies as they evolve. These perspectives define a dynamically managed environment in which processes are repeatedly applied. Long-term preservation requires continuous attention to the properties of the collection, and the ability to verify that those properties are still conserved.

Towards this objective, the NARA TPAP project evaluated assessment criteria for trustworthiness. The original evaluation from RLG/NARA[25] was replaced by the TRAC[26] Trusted Repository Assessment Criteria. The TRAC criteria in turn have been amended by the ISO standardization committee on Mission Operations Information Management Systems[27] repository assessment criteria and form the basis of the ISO 16363 standard. An effort is underway to define the computer actionable assessment criteria, change them into rules that can be enforced by iRODS, and build a preservation environment that periodically validates the criteria. Towards this end, a set of 52 actionable criteria were identified, and then mapped to 20 generic functions that must be supplied by the preservation environment. Many of the criteria require the manipulation of structured information, either through templates that define the required representation information for a record, or through templates that define how state information needs to be organized for an assessment report. The list of functions is given in Table 1.

Number	Preservation function
1	Parse information from document based on a template
2	Create a document based on structure defined by a template
3	Migrate a document between structures defined by two templates
4	Parse required policies from a template specifying rule parameters
5	Generate audit trails for requested operation
6	Parse audit trails to identify specified event/operation
7	Create replicas
8	Synchronize replicas
9	Create and validate checksums
10	Generate event-based notification
11	Associate required metadata with a record
12	Manage a rule base of preservation policies

[25] RLG/NARA audit checklist: http://www.oclc.org/research/activities/repositorycert.html?urlm=22610

[26] TRAC: http://www.crl.edu/archiving-preservation/digital-archives/metrics-assessing-and-certifying-0

[27] ISO Mission Operations Information Management System repository assessment criteria: http://www.crl.edu/archiving-preservation/digital-archives/metrics-assessing-and-certifying/iso16363

Number	Preservation function
13	Apply a required rule to a record
14	Provide unique identifiers for users, records, rules, micro-services
15	Authenticate all users
16	Assign users to roles
17	Authorize all operations based on roles
18	Version policies, micro-services, and state information
19	Manage a staging area for data ingestion
20	Support federation of preservation environments

Table 1. Generic preservation functions.

These functions are all supplied by iRODS, which provides hope that it will be possible to implement a full set of preservation policies. Note that the functions include the parsing of audit trails, as well as the evaluation of current state information. The preservation environment needs to track compliance over time as well as current compliance with policies.

Integrated Preservation Environments

The SHAMAN project integrates the Cheshire3 digital library system and the Multivalent Browser technology with the iRODS data grid. A key part of the integration was the implementation of a micro-service that is capable of executing sub-routines written in the Python programming language. This enabled the iRODS rule engine to control workflows that included Cheshire3 functions. The Multivalent Browser technology provided a means to parse legacy data formats using portable parser technology written in Java.

The integrated system was used to demonstrate the concept of persistent objects. A record is kept in its original data format. When the record is accessed, the legacy format is parsed using the Multivalent Browser technology and transformed into the required display format. This avoids the need to apply transformative migrations to an entire archive, and simplifies long-term administration. The records can persist in their original format. This approach is an intermediate solution between emulation (in which the original display application is kept invariant, but migrated onto new operating systems) and migration (in which the entire archives is transformed to a new data format). The archivist controls the preservation environment, including the iRODS data grid (which provides infrastructure independence), the Multivalent and Cheshire3 technologies (which manage the parsing) and the choice of storage infrastructure.

This approach enables the integration of digital library annotation services with preservation. Annotations can be applied to the records, but are kept as separate metadata associated with each record. The annotations are linked to the record and do not modify the record. An implication is that the annotations are associated with the display of the record and can be mapped to any display choice.

The Cheshire3 technology supports full text indexing. The search indexes are managed as additional information that is linked to and stored alongside the original records. The ingestion of records into the archives can now be differentiated across both client-side workflows (managed by Cheshire3) and server-side workflows (managed by iRODS). The client-side workflows can contain procedures related to content creation and validation, policy ingest, process creation and validation, and infrastructure management and validation. The server-side workflows can contain procedures related to storage administration, validation of assessment criteria, and storage resource-specific enforcement of management policies.

The ability to distribute workflows across ingestion servers as well as archive storage resources makes it possible to optimize management procedures, support scaling of the archives to petabytes of data, and guarantee enforcement of management policies. Bulk operations are supported through iRODS micro-services. Discovery and analysis processing are supported through Cheshire3. These processes include text mining and advanced information retrieval. All policies are processed through the iRODS rule engine. Policies stored in Cheshire3 as RDF/XML documents are searched by an iRODS rule. When an applicable policy is identified, iRODS triggers a Cheshire3 workflow that executes Cheshire3 analysis procedures. This makes it possible to combine both client-side and server-side workflows under the control of the iRODS rule engine. A significant consequence is that no matter which client is used to access the archives, the policies and procedures will be enforced.

All of the iRODS policy hooks can be used to control Cheshire3 workflows. Any operation that manipulates properties of users, storage resources, files, collections and metadata can invoke a Cheshire3 procedure. Cheshire3 provides a mechanism to add metadata as RDF triples and define workflows that will extend the iRODS policies. Two different policy abstractions can be implemented. The iRODS data grid stores its policies in a distributed rule base that can only be modified by the data grid administrator. This implements a highly controlled environment in which the policies are relatively static and stored independently at each storage location. In fact, each storage location can enforce storage-specific policies. With the Cheshire3 implementation, an iRODS rule can be extended to call a Python-based script that can be downloaded to the storage resource. The Python-script invokes the Cheshire3 workflow. This provides an extensibility mechanism that can be enabled by the iRODS data grid administrator, allowing additional users to add procedures to the system. This approach will be important for environments that require multiple institutions or projects to control part of the distributed storage environment. A preservation example is the management of policies that evolve over time. New policy extensions can be tested through the Cheshire3 workflow, while continuing to enforce the original policies. This limits extensions to policies that are compatible with the original preservation policies.

The Cheshire3 workflows are designed to provide distributed information retrieval support and algorithms, and differ from the Hadoop[28]-oriented solutions, which tend to be batch-oriented and not geared towards information access. The current work has focused on addressing a number of outstanding grid-related information retrieval issues, which are as follows:

[28] Hadoop: http://hadoop.apache.org

- We want to preserve the same retrieval performance (precision/recall) while hopefully increasing efficiency (i.e., speed);
- We recognize that very large-scale distribution of resources is (still) a challenge for sub-second retrieval;
- Unlike most other typical grid or cloud processes, information retrieval is potentially less computing intensive and more data intensive;
- In many ways, grid information retrieval replicates the process (and problems) of metasearch or distributed search.

Unified Data Space

The examples above are designed to show that unification of preservation with collection building, digital library services, and data analysis pipelines is feasible. Each environment imposes policies and procedures related to management of record context, with the preservation environment providing the most detailed context. The data management environment shares data across space and time, with future archivists sharing access with current archivists. The context required by the archivist includes the preservation policies and procedures that are used to manage the preservation environment.

The intellectual property in the data management system primarily lies in the set of policies and procedures that are used. The policies can evolve with each stage of the data file cycle, enabling new intellectual property to be defined and added as the user community broadens. Each successive user community can control the context they need to associate with the records, and capture the control mechanisms as intellectual property that justifies their investment in the archives. This is a form of sustainability, with the archives being repurposed for a new use by the implementation of new procedures. The procedures can be controlled by new policies. Both the new procedures and new policies can be added to the preservation environment without having to change the data management framework. The same iRODS framework can be used as generic infrastructure. This approach extends the concept of infrastructure independence to include evolution of the management policies and associated procedures. The preservation environment used in the future can evolve from the preservation environment that is being used today.

The extensibility provided by iRODS was only possible through the management of name spaces for policies, micro-services and state information. These three additional name spaces are used to track new versions of rules, new versions of micro-services, and new versions of state information that is generated by the new micro-services. The ability to evolve the preservation environment is a fundamental preservation principle.

Social processes drive the management of the multiple stages of the data life cycle. Each stage of the data life cycle can correspond to requirements that are generated by a broader community. Each new community has to develop a consensus on how the collection should be managed. This can be viewed either as a process that drives the formation of a common user group, or as a process that socializes the collection, in which the properties of the collection are transformed to meet the requirements of the new community. Socialization of collections corresponds to the repurposing of archives to meet a new set of requirements. The socialization process is viable if the original context (representation information, policies and procedures) can be maintained, and the new context is imported as an augmentation of the original context. Repurposing of collections can be done without destroying the original context.

An example of this approach is used in urban planning. A social consensus is needed to define the information on which planning decisions will be based. The information can be captured as a shared collection, with a context defining the authoritativeness, completeness and authenticity. Policies can be established for what can be admitted into the shared collection. Policies can also be established for the types of allowed analyses that can be applied to the data. As the urban planning group evolves to include additional communities, the policies controlling ingestion and analysis can also evolve. The final form of the collection will represent the social consensus of the entire community. However, the original purpose under which the collection was assembled can still be defined through the original representation information, policies and procedures.

Conclusions

The management of all phases of the data life cycle is now feasible through use of policy-based data management systems. Each phase of the data life cycle is characterized by separate policies and procedures that reflect a consensus of the community that is managing the collection. Each stage of the data life cycle corresponds to a repurposing of the collection for use by a broader community. By tracking the evolution of the policies and procedures through the multiple life cycle stages, the authoritativeness, authenticity, completeness and trustworthiness of the collection can be verified.

A preservation environment can support use by multiple user communities by integration with appropriate access methods. Through the concept of infrastructure independence, new access mechanisms can be added without impacting the enforcement of preservation policies. In the NITRD iRODS demonstration, these capabilities were demonstrated through application of the iRODS technology to collection building, data analysis pipelines, digital libraries and persistent archives.

Acknowledgements

The development of the iRODS data grid was funded by the NSF Office of Cyberinfrastructure OCI-0848296 grant 'NARA Transcontinental Persistent Archive Prototype[29]' (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); NSF OCI-1032732 grant 'SDCI Data Improvement: Improvement and Sustainability of iRODS Data Grid Software for Multi-Disciplinary Community Driven Applications' (2010-2013); and the NSF Cooperative Agreement OCI-094084, 'DataNet Federation Consortium' (2011-2013).

[29] NARA Transcontinental Persistent Archive Prototype: http://www.renci.org/wp-content/pub/techreports/TR-10-04.pdf

References

Moore, R. (2006). Building preservation environments with data grid technology. *American Archivist, 69*, 139-158.

NASA. (2009). NCCS user forum. Retrieved from http://www.nccs.nasa.gov/images/2009_03_24_NCCS_User_Forum.ppt

Phelps, T. & Watry, P. (2005). A no-compromises architecture for digital document preservation. Ninth European Conference on Research and Advanced Technology for Digital Libraries. Vienna, Austria. doi:10.1007/11551362_24

Rajasekar, A., et al. (2003). Storage resource broker: Managing distributed data in a Grid. *Computer Society of India Journal, Special Issue on SAN, 33(4).*

Rajasekar, A., Wan, M., Moore, R. & Schroeder, W. (2006). A prototype rule-based distributed data management. HPDC Workshop on Next Generation Distributed Data Management. Paris, France.