Policy-based Preservation Environments: Policy Composition and Enforcement in iRODS[1]

MIKE C. CONWAY, JEWEL H. WARD, ANTOINE DE TORCY, HAO XU, ARCOT RAJASEKAR, and REAGAN W. MOORE
University of North Carolina at Chapel Hill

Abstract: The management of preservation policies for large digital collections can be an onerous task. Typical policies may control ingest of records, extraction of required provenance information, replication of records, retention and disposition, and validation of assessment criteria. For this study, we chose for this prototype a basic set of policies related to the accession and retention of digital records, and involved in ensuring integrity and authenticity of the records. Policies are presented at an abstract level through graphical user interfaces which are oriented towards archivists. In this paper, we describe a prototype that demonstrates the composition and application of preservation policies.

Problem Statement

Previous empirical research has posited a theory of digital preservation that describes a minimal set of preservation processes required to implement archival management policies, and the minimum set of preservation metadata needed to validate the integrity of a digital repository (Consultative Committee for Space Data Systems 2002), (Moore 2008), (Beagrie 2002). Efforts have been made to enumerate the policies required to produce a trusted digital repository by defining assessment criteria, mapping them to management policies, and identifying those capabilities required to implement the policies (Moore, Rajasekar and Marciano 2007), (Moore 2005), (CRL and OCLC, 2007).

There are a potentially large set of computer actionable rules required to create a trusted digital repository (Moore, Rajasekar, and Wan 2002). The ability to express policy through computer-actionable rules, and the ability to make assertions about the integrity

[1] iRODS, (the integrated Rule-Oriented Data System), is a unique platform that enables the creation of a trusted digital repository through the ability to define and apply preservation policies. The DICE group (Data Intensive Cyber Environments), who develops iRODS, has been working with the National Center for Advanced Systems and Technologies (NCAST) of the National Archives and Records Administration (NARA) on building digital preservation environments based on iRODS. NARA has been particularly interested in preservation policies that enforce integrity, authenticity and chain of custody of their records.

Published by Society of American Archivists, April 2011.

of the preservation environment, can have a significant impact on the cost and effort required to build a trusted digital repository. In order to manage this potentially complex set of policies and attributes, it has become a requirement to develop and implement systems that allow the composition of policies and verification of policy enforcement by the archivist.

Work is proceeding on the top-down definition of policies and attributes that constitute a trusted digital repository (Consultative Committee for Space Data Systems 2009). iRODS (integrated Rule-Oriented Data System) provides the capability both to implement the necessary policies and to maintain the metadata required to manage a secure digital repository (Moore, Rajasekar, and Marciano 2007). Within iRODS, objects can be stored, replicated, and organized into collections representing series of records. In addition, iRODS uses a system of rules to automate the management of data based on various events, such as the ingest of files (Rajasekar et al. 2010). The iRODS system requires the expression of policies as computer actionable rules that control the execution of procedures. The computer actionable rules are stored in a rule base and applied on every interaction with the data management environment. The procedures are implemented as computer executable workflows that are composed by chaining basic functional units (micro-services). The application of policies within iRODS therefore requires a translation from the desired policy into a rule that controls when, where, and what is done to a record.

This project investigated a system to "bridge the gap" between high-level policy, and the enforcement of policy using computer-actionable rules within the policy-driven iRODS repository. Previously, the PLEDGE project identified a goal "to develop its policies and rules engines in such a way that working archivists can interact with the system to specify the policies and maintain them over time" (Smith and Moore 2007). In that spirit, this paper describes a prototype system that allowed specification of policy by the composition of policy templates from components that represent specific preservation actions. The prototype considered key aspects of a user-friendly system intended to support policy development by archivists and other professionals in the practice of digital preservation.

In developing this prototype, we observed the following three principles:

First and foremost, the interfaces must be intuitive and easy to use. The archival interfaces should be organized around use cases and workflows typical to digital repositories. The details of programming rules and micro-services should be hidden from

non-technical users and replaced with interfaces that declaratively specify what the policies should do.

Second, the archival interface should allow archivists to view the status of data objects, and to verify that policies are being applied correctly. Archival interfaces should depict the state of workflows that are part of the policy, and show exceptions and audit information appropriate to the policy and role of the viewer.

A *third* important aspect is that the enforcement of policy must be transparent to the person who is interacting with a digital repository. The applicable policies must be applied regardless of the client used to access the repository.

Method

Our methodology included both the creation of a technical solution in the form of building a prototype and an examination of previous research that explored what types of polices are machine-actionable (CRL and OCLC 2007) . We focused on the technical solution of the study in this paper, and for purposes of scoping the prototype, limited the policy type to those associated with a Submission Information Package (SIP) (Consultative Committee for Space Data Systems 2002).

We created a prototype that allowed us to investigate key aspects of a system that could bridge the gap between high-level policy specification, and computer-actionable rules within a trusted digital repository (Moore, Rajasekar, and Wan 2002). The investigation included the development of interfaces that archivists might use to define and apply policies based on established best practices (Smorul et al. 2004).

We limited the scope of this prototype to include the transfer, validation, and replication of digital records at the point of ingest (Fedora and the Preservation of University Records Project 2006). We specified the transfer and validation at the point where the archive receives the SIP. We limited the ingest function to check whether or not the submitter is authorized to commit the SIP to the archive, whether the SIP contains malicious code, and whether or not the record components are corrupted. Specifically, we focused on providing functionality that authorizes the submitter to provide material to the repository, run a virus scan, calculate and store a checksum, replicate the data, and specify a retention period.

Results and Discussion

As part of the study, we focused on managing the content in the preservation environment, as well as on creating and enforcing policies and legal agreements [14]. As the prototype design proceeded, several roles emerged around which the ingest process was organized. Therefore, we defined the following roles:

- Archive administrator - overall administrator with access to all interface functions.
- Policy administrator - develops and manages repositories of policy templates.
- Collection curator - creates new collections or record series, and applies and customizes policy templates to control the treatment of records in the collection or series.
- Record submitter - uploads content into established collections or record series.

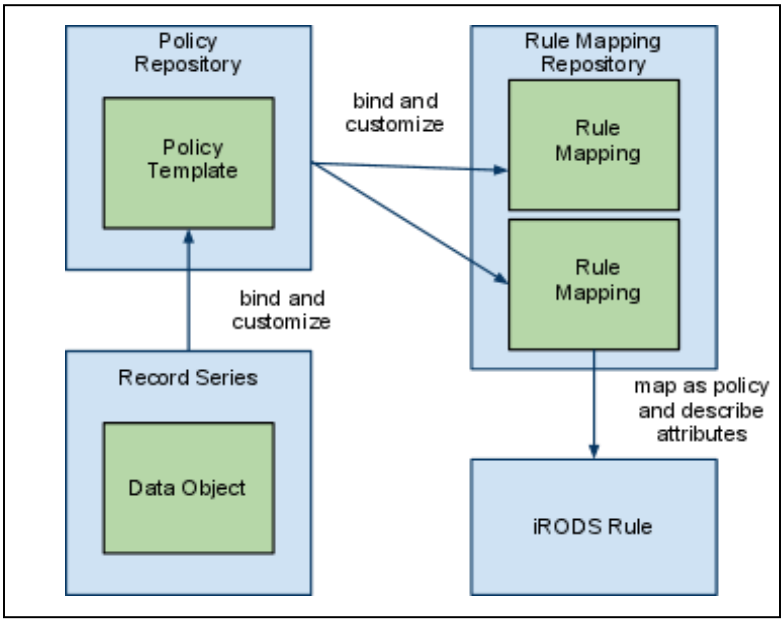
Based on these roles, a set of high-level components was identified:

- An "archive manager" component that can track repositories of computer-actionable rule mappings, and policy templates.
- A "rule mapping" of computer actionable rules that describes an iRODS rule through "plain language" descriptions of a policy element, and describes the attributes that tailor the actions of the rule such that these attributes may be displayed and customized via a user interface. These mappings can be placed into a rule mapping repository.
- A description of a "policy template". In part, these templates are composed by selecting rule mappings from a repository, and setting of default properties based on the attributes described in the rule mapping. The policy templates may be placed in a policy repository.
- A "record series" creation component that can define a new collection of records, such that they are bound to a selected policy. At the time of binding, a policy template is selected from a policy repository, and this policy is displayed for customization.
- A "policy resolution" process where operations on a data object or collection with a bound policy are recognized, and the controlling policy is discovered and activated.

The policy binding framework (Figure 1) describes the proposed architecture at a high level. In order to encapsulate the necessary medatada about the policy, mechanisms to represent the policy template, and the binding of policy templates to record series were considered. In this case, the application stored the various mapping and template files as XML within a set of standard "shadow directories" in the archive. Further, the

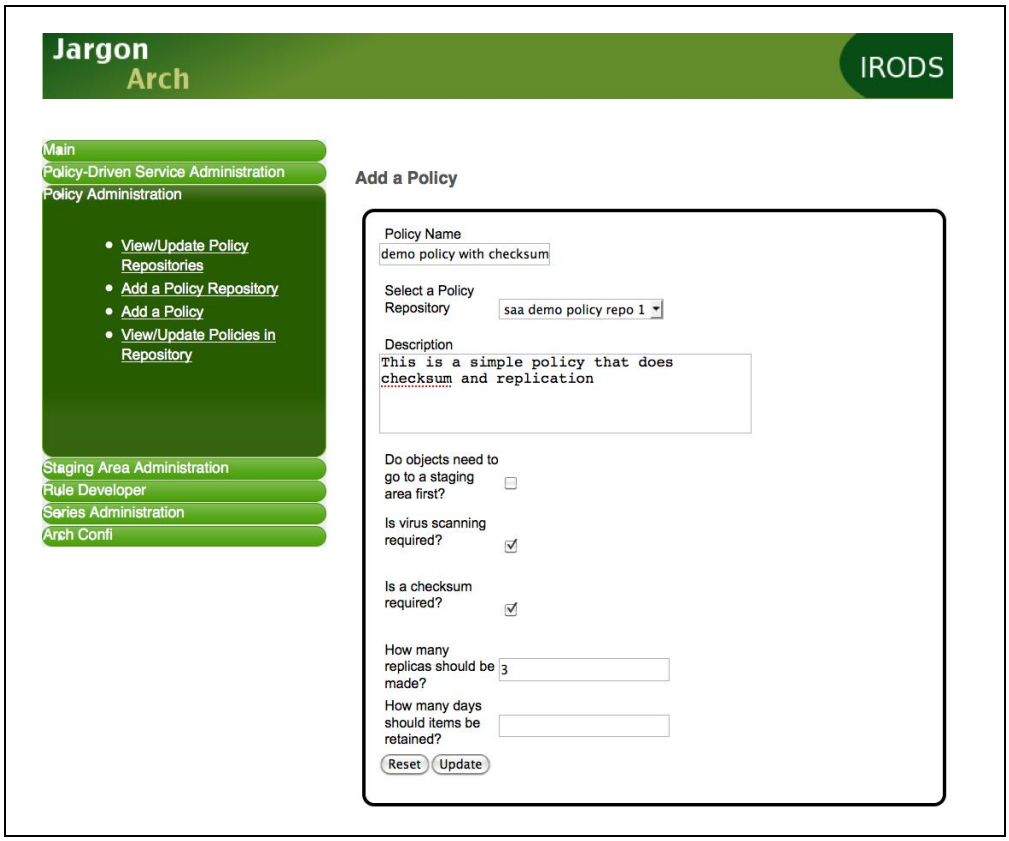
application used iRODS user-defined metadata expressed as Attribute, Value, Unit (AVU) triples to link the various parts of the system together, and to record customizations of the various elements. These AVU mappings are implied in the "bind and customize" links in
Figure 1.

Figure 1. Policy binding framework



The interface for policy definition and policy binding in our prototype is called Arch (Figure 2). This web application was able to utilize the above described framework of mappings, repositories, and bindings between a policy and a record series, resulting in the dynamic application of iRODS rules based on policy definitions.

Figure 2. Arch main menu screen shot

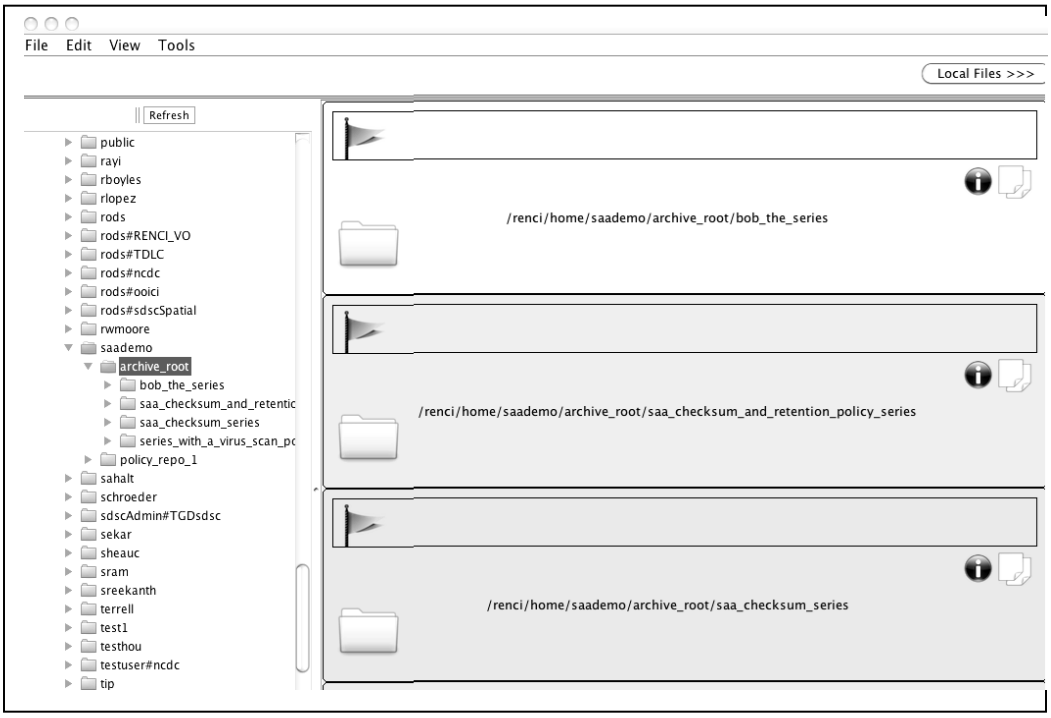


When a user selects a policy to be applied to a series, Arch adds metadata to the corresponding collection in iRODS. The policy is bound to the series in the form of metadata attributes referencing an existing policy and specifying parameters for that policy.

A rule engine in iRODS controls the enforcement of preservation policies, taking on the function of the "policy resolution" component. Each rule is comprised of a condition and a workflow made of micro-services, which are commands that the iRODS server can invoke and that are each designed to do a given task. Policy components available when creating a policy correspond to operations that can be performed by existing iRODS micro-services. An accession policy action thus consists of a sequence of micro-services executed upon ingest of a record. Every time a new record is ingested into a series, iRODS looks at the metadata associated with that series, and if policy related metadata attributes are present a sequence of micro-services representing the policy's actions is executed.

As the prototype was scoped to the ingest function, a complementary data transfer client was developed. The ingest client was called "iDrop", and can be described as a series viewer primarily designed to allow easy and seamless data ingest through drag and drop actions (Figure 3). A transfer manager was implemented to manage and monitor large queues of transfers, and to handle network interruptions.

Figure 3.- iDrop screenshot



This client was developed so to be "policy aware". Each micro-service sequence designed to apply a policy action to an object also updates metadata attributes representing the object's status. For this prototype, specific icons have been made and integrated with iDrop to represent application of virus scan and checksum policies. They automatically appear when the policies have been applied. For replication and retention period policies one can see the status of an object in regards to these policies by clicking on the metadata icon next to the object's name in iDrop. The series browser in iDrop therefore displays collections and objects in a way that reflects compliance (or not) to their associated policies.

This project demonstrated a system to "bridge the gap" between the high-level policy definition, and the enforcement of policy using the policy-driven iRODS repository.

The ability to declaratively build and apply policies in a digital repository were demonstrated at the National Archives and Records Administration (NARA) through the Arch and iDrop interfaces to the Transcontinental Persistent Archives Prototype (TPAP) test bed. The significant components of the demonstration were:

- *Ease of use of the interfaces.* The Arch interface was designed to present policies to the user in terms of natural language constructs. The user entered the minimal amount of information needed to establish the policy input parameters. The user then defined which policies would be applied to a record series.

- *Automation of data transfers.* The iDrop interface manages the transmission of data from a remote computer into the data grid, which requires monitoring the success of the transmission, maintaining a log of all attempted data transfers, tracking errors, and presenting a report to the user of what was accomplished. The interface can be extended to automate recovery from standard errors. This greatly decreases the effort needed to ingest large numbers of records.

- *Automated policy detection.* A generic policy for processing the ingest of records was developed. The generic policy queried the collection under which the record was being saved, extracted the set of policy input parameters from the collection attributes, decided which rule needed to be executed, and automatically invoked the correct procedures. The implication is that the invocation of policies was reified in terms of the policy input parameters. This minimizes the number of rules that need to be implemented, and characterizes each policy in terms of the input parameters.

- *Standard policy sets.* A policy repository was created within the TPAP test bed to hold an XML-based characterization of each policy. The process decoupled the specification of a policy from the association of the policy with a record series. Thus a policy could be defined, and then be independently applied to multiple record series, each with a different set of input parameters.

- *Standard rule sets.* Each policy was composed from multiple rules that in turn were managed within a rule repository. Each rule was characterized as an ASCII file and stored in a rule repository within the TPAP test bed. Thus new rules could be added and new versions of rules could be created.

Findings and Conclusion

This prototype shows that computer actionable rules can automatically apply a set of policies and enforce desired preservation properties. At this point however the creation of preservation policies in iRODS still requires significant knowledge of the underlying system.

Within the scope of the prototype, we demonstrated how high-level policies could be mapped to low-level implementations in the iRODS rule language. The prototype also showed that the manipulation of the high-level policies could be done via graphical user interfaces.

The representation of policies in the prototype was closely tied to existing iRODS rule engine capabilities. The development or adoption of a cross-repository generic policy expression language was not in the scope of the prototype. Further specification of a generic policy language that can express the policy set is an important effort. Groups such as the W3C Policy Languages Interest Group have identified several interesting policy languages and use cases. We feel that the idea of policy composition, binding, and customization via graphical interfaces is an interesting perspective on the larger question of the translation of high-level policy to actionable rules.

The prototype concentrated on one type of rule, atomic rules, which are "those which occur on execution of a specific event (in the event-condition-action model) and are most often evaluated at the item level on each execution of a related operation" (Smith and Moore 2007). Since the ingest rules were limited to a subset of the possible events that could occur within iRODS, it was a straight-forward mapping. iRODS itself presents sixty-nine published event 'hooks' where iRODS rules are automatically enforced. The event hooks cover the initiation and completion of various operations within iRODS, such as the ingest of a data object, or the creation of a collection. An important conceptual exercise to further define a bridge between high-level policy and computer-actionable rules within iRODS would be a mapping of events in an archive to the policies that pertain to that particular event at an abstract level. This will be most important in any elaboration of a "policy resolution" component that recognizes such an event, resolves the applicable policy, and applies the policy at the correct time within iRODS. In the design of the prototype, this sort of mapping was also discussed in terms of steps in the archival process, such as accession and arrangement. Limiting the prototype to a subset of the ingest process allowed us to proceed in development, but the lack of such conceptual event mapping is a significant limitation to creating a generalizable solution.

The design of a policy-driven repository with the qualities suggested by this prototype rests on the required suite of rules and associated metadata attributes described in the 'theory of digital preservation' (Moore 2008). The definition of these rules and metadata attributes remains the fundamental question. It has been shown that such mapping of policies and attributes is feasible, and this process has been illustrated in some detail (Smith and Moore 2007). The Arch and iDrop prototype systems show potential interfaces and mechanisms to implement high-level policy handling using iRODS. Such a system could bridge "the gap" between high-level policy and computer-actionable rules, and present tools and graphical interfaces to archivists to configure and monitor the application of policies in a digital repository.

References

Beagrie, Neil. et al. 2002. *Trusted Digital Repositories: Attributes and Responsibilities.* RLG-OCLC Report. http://www.rlg.org/longterm/repositories.pdf.

Consultative Committee for Space Data Systems. 2009. *Audit and Certification of Trustworthy Digital Repositories* (CCSDS 652.0-R-1). http://wiki.digitalrepositoryauditandcertification.org/pub/Main/WebHome/652x0r1candidate-update-typoscorrected.doc.

Consultative Committee for Space Data Systems. 2002. *Reference model for an Open Archival Information System (OAIS) (CCSDS 650.0-B-1).* Washington, DC: National Aeronautics and Space Administration (NASA).

CRL and OCLC. 2007. *Trustworthy Repositories Audit & Certification: Criteria and Checklist*. The Center for Research Libraries (CRL) and the Online Computer Library Center (OCLC). http://www.crl.edu/sites/default/files/attachments/pages/trac_0.pdf.

DICE. *The iRODS Rule System*. https://www.irods.org/index.php/The_iRODS_Rule_System.

Moore, Reagan. 2005. "Persistent collections," in S.H. Kostow and S. Subramaniam (Eds.), *Databasing the Brain: from Data to Knowledge (Neuroinformatics).* 69-82. Hoboken, NJ: John Wiley and Sons.

Moore, Reagan. 2008. “Towards a Theory of Digital Preservation”. *International Journal of Digital Curation* 3, no. 1. 63-75.

Moore, Reagan, Arcot Rajasekar and Richard Marciano. 2007. “Implementing Trusted Digital Repositories”. *Proceedings of the DigCCurr2007 International Symposium in Digital Curation*, University of North Carolina - Chapel Hill.

Moore, Reagan W., Arcot Rajasekar, and Mike Wan. 2002. *Policy Sets As Standards for Preservation*. http://egdpm.nist.gov/workshop/papers/02_02_NIST-irods.doc.

Rajasekar, Arcot, Reagan Moore, Chien-yi Hou, Christopher A. Lee, Richard Marciano, Antoine de Torcy, Michael Wan, Wayne Schroeder, Sheau-Yen Chen, Lucas Gilbert, Paul Tooby, Bing Zhu, and Gary Marchionini (ed). 2010. *iRODS Primer: Integrated Rule-Oriented Data System Synthesis Lectures on Information Concepts, Retrieval, and Services*. Morgan & Claypool.

Smith, MacKenzie, & Reagan W. Moore. 2007. “Digital Archive Policies and Trusted Digital Repositories”. *The International Journal of Digital Curation* Issue 1, Volume 2 (2007): 92-101.

Smorul, M., Joseph JaJa, Yang Wang and Fritz McCall. 2004. *PAWN: Producer-Archive Workflow Network in Support of Digital Preservation*. CS-TR-4607, UMIACS-TR-2004-49. http://www-lb.cs.umd.edu/Library/TRs/CS-TR-4607/CS-TR-4607.pdf.

Tibbo, Helen R. 2003. *On the Nature and Importance of Archiving in the Digital Age*. *Advances in Computers* 57 (2003): 2-20.

Tufts University and Yale University. 2006. *Fedora and the Preservation of University Records Project. 2.1 Ingest Guide*, Version 1.0. http://repository01.lib.tufts.edu:8080/fedora/get/tufts:UA069.004.001.00006/bdef:TuftsPDF/getPDF.