Journal of
open research software

Moore, R et al 2014 Extensible Generic Data Management Software. *Journal of Open Research Software,* 2(1): e9, pp. 1-5, DOI: http://dx.doi.org/10.5334/jors.ap

ISSUES IN RESEARCH SOFTWARE

Extensible Generic Data Management Software

Reagan Moore*, Arcot Rajasekar* and Hao Xu*

The integrated Rule Oriented Data System (iRODS) provides an excellent use case for exploring sustainable software development. The iRODS data grid software is the product of a 20-year development effort that relied on funding from multiple federal agencies. Each agency needed infrastructure to support multiple scientific disciplines. Each discipline potentially required multiple types of data management (sharing, publication, preservation, and processing). The iRODS software was developed as generic infrastructure that could be tuned to implement the desired data support mechanisms. Active engagement of the user communities was encouraged through open source collaborative development. Based on the successful application of the software across the multiple disciplines, we make three claims about sustainable software development: 1) that the success of the software is dependent on having a highly extensible architecture; 2) that extensible software enables infrastructure re-use across domains; and 3) that consortium based development enables long-term support for software infrastructure.

Keywords: data grid; policy-based data management

Introduction

The Data Intensive Cyber Environments (DICE) group has collaborated with more than 25 science and engineering domains on the design of open source distributed data management systems, known as iRODS, the integrated Rule Oriented Data System [1]. The software is middleware that organizes distributed data into shareable collections, while enforcing management policies. The software is used to manage data for project collections, collaboration environments, digital libraries, and data archives. A partial list of domains and projects that build on the iRODS extensible generic data management infrastructure includes:

Biology	
Cognitive Science	Temporal Dynamics of Learning Center
Human genome	Broad Institute, Wellcome Trust Sanger Institute, NGS
Medicine	Sick Kids Hospital
Neuroscience	International Neuroinformatics Coordinating Facility
Plant genome	the iPlant Collaborative
Phylogenetics	Phylogenetics at CC IN2P3
Computer Science	
Network research	GENI experimental network
Earth Sciences	
Atmospheric science	NASA Langley Atmospheric Sciences Center
Climate	NOAA National Climatic Data Center NASA Center for Climate Simulations
Ecology	CEED Caveat Emptor Ecological Data
Hydrology	Institute for the Environment, UNC-CH; Hydroshare
Oceanography	Ocean Observatories Initiative
Seismology	Southern California Earthquake Center
Engineering	
Education repository	CIBER-U
Physics	
Astrophysics	Auger supernova search
Cosmic Ray	AMS experiment on the International Space Station
Dark Matter Physics	Edelweiss II
High Energy Physics	BaBar / Stanford Linear Accelerator
Neutrino Physics	T2K and dChooz neutrino experiments
Optical Astronomy	National Optical Astronomy Observatory
Particle Physics	Indra multi-detector collaboration at IN2P3
Quantum Chromodynamics	IN2P3
Radio Astronomy	Cyber Square Kilometer Array, TREND, BAOradio
Social Science	Odum, TerraPop

* University of North Carolina at Chapel Hill, United States
rwmoore@renci.org

The applications range from management of small gigabyte-sized collections with a few thousand files, to sharing of multi-petabyte sized collections with hundreds of millions of files that are distributed internationally. The number of users may be less than ten persons, to more than 10,000 users. The collections may be distributed across multiple institutions and multiple continents. The applications include institutional repositories for managing reference collections, regional data grids for data sharing, national data grids, national digital libraries, national archives, and international collaborations. The wide range of applications required development of an architecture that could be extended to support the specific requirements of each community. The combination of extensibility mechanisms on top of generic infrastructure was essential for adoption of the iRODS software by a large number of communities, and for the sustained interest in the software.

Specific applications of the iRODS data grid include: the iPlant Collaborative [2] that has 13,000 users and more than 100 Terabytes of data organized by small project teams that share tightly controlled collections; the BaBar High Energy Physics project [3] which has moved more than 2 Petabytes of data from the Stanford Linear Accelerator in Palo Alto, California to Lyon, France; and the NOAA National Climatic Data Center that has built a staging area to manage the ingestion of all submitted climate records into a preservation environment [4].

Each application has unique semantics, data formats, types of data, analysis procedures, management policies, descriptive metadata, and hardware systems (including unique network access protocols). Each domain has existing infrastructure that manages legacy data, provides analysis services, and serves as an authoritative resource for domain knowledge. Each domain may organize data in a collection, or share data within a data grid, or publish data in a digital library, or preserve data in an archive. Given these diverse requirements, sustainable software requires extensibility mechanisms that enable generic infrastructure to be applied by each domain.

Capturing Domain Knowledge

The approach taken in the DICE group has been to build middleware that captures domain knowledge. This may be knowledge that is needed to translate from access protocols required by domain resources to the client protocols desired by researchers. Or it may be knowledge about the required descriptive metadata or procedures for manipulating the data formats. Or it may be knowledge needed to provide a unified view across heterogeneous production systems. A unified view constitutes a collaboration environment through which researchers can access existing resources, share data, information, and knowledge, and manage their research data.

The iRODS data grid uses a distributed rule engine to apply policies and procedures that articulate domain knowledge. The policies are expressed as computer actionable rules managed in a rule base and applied by a distributed rule engine. The procedures are expressed as workflows that are composed by chaining together basic functions called micro-services. The application of iRODS to a new domain is accomplished by adding domain specific policies and procedures. For example, the iPlant Collaborative applies procedures for processing genomic data files, while the BaBar project implemented rules for automating replication of data.

The iRODS data grid implements a collaboration environment that incorporates infrastructure independence, enabling migration of data across hardware and software systems. The virtualization environments implemented by iRODS simplified the application of the technology to diverse scientific domains.

Essential components of domain knowledge are applied through mechanisms that enable infrastructure independence. Infrastructure independence is implemented through interoperability mechanisms that enable use of multiple types of technology within the collaboration environment. The interoperability mechanisms can be categorized through the types of data manipulation operations that each domain performs upon the following eight name spaces:

1.	Users	(group formation, authorization, authentication, audit)
2.	Resources	(storage interaction, remote application execution, queuing)
3.	Data Objects	(replication, versioning, distribution, streaming, transport)
4.	Collections	(access controls, archiving, soft links, registration)
5.	Metadata	(schema, ontologies, vocabularies)
6.	Policies	(enforcement points, automation, versioning)
7.	Procedures	(workflow provenance, re-execution, versioning, sharing)
8.	Events	(access, usage, changes)

The eight name spaces have been implemented in the iRODS integrated Rule Oriented Data System, along with the interoperability mechanisms that execute the processes needed to apply the desired operations across existing hardware and software systems [5]. Note that the approach has to enforce management policies across administrative domains, provide a single sign-on environment for users, enable re-use of existing data collections, enable processing both at the place where data are stored and at compute engines, and maintain a consistent and persistent set of provenance, descriptive, and administrative metadata. These capabilities ensure that the software will be able to incorporate new technology as it becomes available. This in turn ensures that the software will continue to be useful in the future, improving the long-term support prospects.

Extensibility Mechanisms

Extensibility mechanisms enable forms of knowledge capture. The knowledge of how to access a remote system or execute a procedure is captured within a procedure that is applied by interoperability mechanisms. In particular, sustainable software provides the interoperability mechanisms needed to incorporate new technology. The approach taken for building sustainable software is best illustrated through examples of requirements from user communities, and through a description of the generic knowledge capture mechanisms that were implemented to meet each requirement.

A dominant requirement has been the ability to capture management knowledge in computer actionable rules. A driving use case from the UK e-Science data grid was a request for the ability to create a collection in which files were permanently managed and could never be deleted by anyone. But at the same time, the ability to manage a collection in which administrators could replace corrupted files was desired, along with the ability for users to update their own files in their own collections. This implied the need to manage at least three different consistency constraints on data deletion within the same data management system (no deletion allowed, deletion by administrator, deletion by file owner).

The DICE group developed the iRODS policy-based system to extract knowledge about management policies from the software, and apply the knowledge through computer actionable rules stored in a rule base. Effectively, every software encoded consistency constraint was replaced by a policy-enforcement-point. Actions by clients were trapped at the policy-enforcement-points. By searching the rule base, an appropriate rule could then be identified which controlled the execution of a workflow that enforced the required management policy. This meant that the knowledge needed to manage the system could be captured in computer actionable rules. The system was no longer restricted to managing files and static representations of information. Instead, a data management system could use rules that controlled the behavior of the system and administrators could dynamically change the rules in a rule base. It became possible to use generic infrastructure to implement archives, digital libraries, data grids for sharing data, project collections, and processing pipelines simply by changing the rules and procedures enforced by the system.

Within iRODS, policies can be enforced for preservation (authenticity, integrity, chain of custody, preservation of the original arrangement of files in a record series, retention, disposition); or for data publication in a digital library (descriptive metadata annotation, arrangement of files in a collection hierarchy, creation of presentation versions such as image thumbnails); or for sharing in a data grid (access controls, distribution, caching); or for reproducible data driven research in a processing pipeline (workflow procedures, workflow provenance, workflow re-execution); or for validating assessment criteria (repository trustworthiness, compliance with regulations).

A second form of knowledge capture is the management of provenance and descriptive metadata. Each science and engineering domain uses different descriptive terms. The iRODS data management system uses schema indirection to enable each community to apply their desired metadata. In essence, descriptive and provenance information are turned into triplets: metadata attribute name; metadata attribute value; and metadata attribute comment. This approach makes it possible for each community to independently specify the information context associated with their collections.

A third form of knowledge capture is automated capture and management of workflow provenance information. Within iRODS, a workflow collection can be associated with a file that contains a workflow written in a workflow language. The output from each execution of the workflow can be captured, along with the input files. This enables reproducible data-driven research through the sharing of the workflow, the input files, and the output files. A researcher can share a workflow analysis with another researcher, or make the workflow publicly accessible. It then becomes possible to re-execute an analysis done by another scientist, modify the input files, re-run the analysis and compare results.

A fourth form of knowledge capture is the automated management of data streams. Within iRODS, an archive collection can be associated with a data stream. Data that are deposited into the archive collection are automatically indexed based on a stream time parameter. Data within a specified time interval can then be retrieved in a single data stream. The data grid automatically does the required sub-setting of files for the start and end of the stream, and composites the intermediate files into the requested data stream.

A fifth form of knowledge encapsulation is through basic functions (micro-services) that can execute the network protocol needed to interact with an external data repository. The micro-service manages the communication, and caches the retrieved data within the collaboration environment. This enables a researcher to link external data sets into a collaboration environment, apply analyses, and manage results while maintaining control over the input files.

Each of these types of knowledge capture is an example of re-use of data grid infrastructure to support a new science and engineering domain. Based on experience with 25 domains, three types of interoperability mechanisms are needed to apply domain knowledge and enable software re-use:

1. Policies that control the execution of procedures, management of data, and verification of assessment criteria.
2. Micro-services that manage interactions with external network protocols, encapsulate specific operations, and encapsulate workflow operators (conditional tests, loops, arithmetic).
3. Middleware servers that apply data grid operations at remote storage locations (Posix I/O commands, staging, archiving) through storage-specific drivers.

Using these three interoperability mechanisms, the iRODS software has been successfully re-used for institutional repositories, regional data grids, national data grids and national libraries, and international collaborations.

Re-use of software is also facilitated through the ability to dynamically change data management system components. In the iRODS Consortium software release v4.0, each of the interoperability mechanisms is dynamically pluggable [6]. It is possible to add a new policy, add a new micro-service, and add a new storage driver while the system is running. This makes it possible to evolve production environments. New types of resources can be added without having to stop the production system.

Consortium-based Development

Even if the system architecture enables re-use by new communities, a central component of sustainability is institutional support for each new community. The expectation is that the lifetime of the institutional commitment will be as long as the lifetime of the technology. Since the iRODS software was developed through research projects funded by United States federal agencies, long-term sustainability required the creation of an appropriate institutional support mechanism.

The community that is most interested in the continued evolution of the technology is the user community that applies the software as a critical component of their infrastructure. For the iRODS data grid, a community of users has been identified who rely upon the software to maintain their intellectual property (data records, documents, workflows, operational procedures, management policies, and assessment criteria). The user community is being organized into an iRODS Consortium, which provides long-term software support and feature development. Each group that joins the consortium makes a funding commitment. In exchange, the consortium provides consulting support, accepts input from each group on new features that are needed, and provides new releases with bug fixes and feature enhancements. To promote the success of the consortium, the Data Intensive Cyber-Environments group issued its last release of the iRODS data grid, version 3.3.1, on February 24, 2014. The first release of the iRODS data grid by the iRODS Consortium was version 4.0, released in March 2014.

This sustainability model addresses multiple challenges:

- Replaces an academic support model with a consortium support model. The groups using the technology are now in control of the evolution of their critical infrastructure components.
- Provides a sustainable funding model that is driven by actual use of the technology.
- Provides support for academic users, research institutions, and commercial companies.
- Provides a way for groups that offer service contracts for consulting support to interact with the user community.
- Enables collaborative development, with contributions to open source software provided by community members.
- Enables communities to separately control intellectual property. In the case of iRODS, the intellectual property is captured as policies and procedures that enforce local management decisions. A community can implement policies that are unique to their organization and manage the policies and procedures independently of the iRODS Consortium generic infrastructure. This makes it possible for a commercial enterprise to participate in open source software development.

Summary

Sustainable data management systems implement an architecture that enables the encapsulation of intellectual property into modules that can be plugged into generic infrastructure. Each user community can build upon a sustainable generic core, while implementing the community-specific mechanisms needed to automate enforcement of management policies, automate administrative tasks such as data migration, automate validation of assessment criteria, capture knowledge (processes) associated with creating derived data products, capture knowledge (communication protocols) needed to interact with remote systems, and automate processing of data within workflow pipelines. The automation of these tasks corresponds to the creation of knowledge procedures that can be applied by the generic policy-based data management system. Extensible data management systems provide the interoperability mechanisms that enable integration with new technologies, federation across institutional repositories, and creation of national-level and international-level collaborations. The ability to separate intellectual property from generic infrastructure enables academia, federal agencies, and commercial companies to collaborate on sustaining the core software infrastructure.

Acknowledgements

The conclusions in this paper have been funded by the NSF Office of Cyberinfrastructure grant OCI-1032732 “SDCI Data Improvement: Improvement and Sustainability of iRODS Data Grid Software for Multi-Disciplinary Community Driven Application” and the OCI-0940841 grant, “DataNet Federation Consortium”.

References

1. **Rajasekar, A, Wan, M, Moore, R** and **Schroeder, W A** 2006 Prototype Rule-based Distributed Data Management System. In: HPDC workshop on Next Generation Distributed Data Management, Paris.
2. **The iPlant Collaborative** Available at: https://www.iplantcollaborative.org
3. **Nief, J-Y, Kroeger, W** and **Hasan, A** 2005 BaBar data distribution using the Storage Resource Broker (SRB). In: HEPiX conference. SLAC.
4. **Hall, A** NOAA’s National Climatic Data Center’s Plan for Reprocessing Large Datasets. Available at: http://storageconference.org/2011/Presentations/MSST/6.Hall.pdf
5. **Rajasekar, A, 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.
6. **iRODS Consortium** Available at: http://www.irods-consortium.org

How to cite this article: Moore, R, Rajasekar, A and Xu, H 2014 Extensible Generic Data Management Software. *Journal of Open Research Software,* 2(1): e9, pp. 1-5, DOI: http://dx.doi.org/10.5334/jors.ap

Published: 9 July 2014

Copyright: © 2014 The Author(s). This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License (CC-BY 3.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited. See http://creativecommons.org/licenses/by/3.0/.

]u[*Journal of Open Research Software* is a peer-reviewed open access journal published by Ubiquity Press.

OPEN ACCESS