Geo-Science Use Cases for Data-Driven Science

Reagan Moore (University of North Carolina at Chapel Hill
rwmoore@renci.org
216 Manning Hall, CB #3360
Chapel Hill, NC 27599-3360
Telephone: 919 962 9548
Fax: 919 962 8071)
Ilkay Altintas (University of California, San Diego)
Daniel Crawl (University of California, San Diego)
Lawrence Band (University of North Carolina at Chapel Hill)
Jong Lee (National Center for Supercomputing Applications)
Yong Liu (National Center for Supercomputing Applications, now Microsoft)
Brian Miles (University of North Carolina at Chapel Hill)
Arcot Rajasekar (University of North Carolina at Chapel Hill)
Antoine de Torcy (University of North Carolina at Chapel Hill)
Ilya Zaslavsky (University of California, San Diego)

Abstract:

Data-driven science is based on the ability to answer research questions through the analysis of collections of data sets. In the NSF EarthCube Initiative Layered Architecture Concept Award, use cases have been defined and implemented that require the analysis of data from multiple federal repositories. The use cases include an analysis of hypoxia in the Gulf of Mexico, analysis of drought in Texas, and an ecohydrology analysis of the cycling of water, carbon, and nitrogen through coevolved climate-soil-vegetation systems. A loosely coupled federation architecture is illustrated that enables the implementation of each use case, the retrieval of appropriate data sets, and the preservation of the research workflows. One of the goals of this infrastructure is reproducible science, the ability to share not only research results, but also the input files and workflow analyses that generated the results.

Acknowledgements:

This work was supported by NSF grant 1239678 “EAGER: Collaborative Research: Interoperability Testbed-Assessing a Layered Architecture for Integration of Existing Capabilities”, NSF grant 0940841 “DataNet Federation Consortium”, NSF grant #1238420, “EAGER: Readiness of Disciplinary Data Systems for Cross-Domain Interoperability within a Standards-Based EarthCube Reference Framework”, and NSF grant 1032732, “SDCI Data Improvement: Improvement and Sustainability of iRODS Data Grid Software for Multi-Disciplinary Community Driven Application”.

1. Introduction:

The NSF EarthCube initiative has the goal of defining and implementing a data cyberinfrastructure architecture that will support reproducible science for Geoscience researchers. The cyberinfrastructure is tasked with integration of community resources,
support for collaborative research, and preservation of research results for all Geoscience research initiatives, including atmosphere, ecology, climatology, environment, geochemistry, geology, glaciology, hydrology, meteorology, oceanography, seismology, and soils. Each sub-discipline is expected to generate unique requirements for data repositories and analysis workflows that need to be supported. Examples range from support for a wide range of data types (gridded data, point data, relational data, streaming data), to support for domain specific semantics and data formats, to support for web services for desired analyses. Each sub-discipline has existing community resources that represent data repositories, information catalogs, knowledge ontologies, and service registries.

The researcher requirements are captured in use cases that illustrate the challenges that confront research initiatives. Fortunately the use cases have common properties that can be supported by generic infrastructure. Many communities require the ability to analyze data records that are stored in geographically distributed federal repositories. The data records are typically retrieved and cached within a research environment for analysis through workflows. The analyses may require integration with external web services and the movement of data back and forth to external servers. The workflow analyses may be shared with researchers and groups located at different institutions. The results may be published after organization into a collection with provenance and descriptive metadata. The research collaboration may decide to formally create a reference collection for use by the rest of the discipline.

Scientific data collections are now being created that contain hundreds of millions of files and petabytes of data generated through complex scientific workflows. The conduct of the scientific endeavor requires data management, workflow management, capture of the context and system environment in which the data are created, and the management of the link between workflows and input and output files to track provenance. The inherent complexity in the creation of data products through orchestration of multiple workflows will require mechanisms to automate administrative and application tasks, to enforce management of the scientific workflow environment, and to capture the contextual information as part of the data creation process.

The creation of support mechanisms for this wide range of research requirements has been proceeding independently within each Geoscience sub-discipline. The expectation is that the integration of the community resources within a unified cyberinfrastructure will improve the ability of geoscientists to conduct research, lead to multi-disciplinary collaborative research, and enable future research initiatives to build upon current research results. The EarthCube initiative has identified a loosely coupled federation architecture as an appropriate mechanism for the integration of community resources. This enables the re-use of existing resources, and provides mechanisms for interoperability between the resources while enabling research collaborations.

The NSF Layered Architecture Concept Award\(^1\) uses the technology being applied within the NSF DataNet Federation Consortium (DFC)\(^2\) to implement prototype data

---

\(^1\) NSF EarthCube Layered Architecture concept award, [http://earthcube.ning.com/group/layered-architecture-concept-award](http://earthcube.ning.com/group/layered-architecture-concept-award)
cyberinfrastructure for Geoscience. The DFC uses policy-based data grids to manage interoperability across community resources, enforce technical governance policies, and manage registration and sharing of analysis workflows. This paper describes the DFC architecture in Section 2, the application of the DFC architecture to ecohydrology workflows in Section 3, the application to river flow models of drought in Texas in Section 4, and the application to studies of hypoxia in the Gulf of Mexico in Section 5. In Section 6, the approach for managing the scientific workflow environment is described, and a summary is presented in Section 7.

2. Datnet Federation Consortium Architecture
Policy-based data management systems provide the ability to compose a logical collection of data and workflows from distributed resources that are linked by a network. The DFC uses policy-based systems to implement shared collections, manage the evolution of policies that control assertions about the shared collection properties, express policies as computer actionable rules, control procedures that manipulate the data, and apply the procedures through computer executable workflows. These capabilities form the basic requirements for a collaboration environment. Researchers can share data, share workflows, use metadata to describe provenance information, add descriptive metadata to data, and use policies to control access and manage the collection. The organizers of the shared collection can automate administrative functions, such as creation of replicas or migration to new storage resources, and can periodically validate assessment criteria. By changing the policies that govern the properties of the shared collection, the stages of the data life cycle can be managed. Collections can be initially assembled within a data sharing environment, then analyzed in a data processing pipeline, published in a digital library, and preserved as a reference collection in an archive. Policies can be applied that represent a consensus on how the project collection should be managed at each stage of the collection life cycle. The original collection policies can be modified to enable publication of the collection, or can be modified to enforce preservation within an archive.

The technology that is used to implement a policy-based data management system is the integrated Rule Oriented Data System\(^3\) (iRODS) (Rajasekar, 2010). The iRODS system is a data grid that provides the virtualization mechanisms needed to enable creation of a collection that spans multiple types of storage systems and multiple administrative domains. The approach is based on the identification of the types of objects that will be managed within the collection, and the creation of logical name spaces that uniquely identify each object. For each name space, associated operations are provided that manipulate entities defined by the logical name space. For each set of operations, virtualization mechanisms are defined to enable application of the operations across multiple types of storage systems and operating systems.

Table 1 illustrates the seven logical name spaces supported within iRODS. Persistent names can be applied to Users, Objects, Collections, State Information, Storage Systems, Policies, and Procedures. The persistent names are assigned independently of the name

---


\(^3\) iRODS integrated Rule Oriented Data System, http://www.irods.org
used within the local administrative domain. Thus the persistent names are referred to as logical names to differentiate them from the physical names used by the local administrator. A mapping is maintained between the logical names and the local names. This makes it possible to group entities within the logical name space, and manage the entities as if they were located in a virtual collection.

Typical operations performed upon the Users name space include authentication, authorization, and organization of groups of users. The Objects name space can be used to manage files, workflows, containers, and soft links. Operations performed upon the Objects name space include partial I/O on files, execution of workflows, and replication. Attributes can be associated with a name space to manage persistent state information that is stored as metadata. This includes both provenance information and descriptive information. Groups of resources can be identified to enable collective operations such as load leveling and fault tolerance. Within the Policy name space, policies are cast as computer actionable rules that control the execution of procedures that are implemented by chaining basic functions into workflows.

The virtualization mechanisms for each name space enable interaction with multiple types of identity and authentication systems, the mapping of POSIX I/O operations to the protocol required by a particular storage or operating system, and the manipulation of metadata in a preferred relational database. Data grids implement two levels of virtualization. Actions that are requested by a client are mapped to standard I/O operations by the middleware. Procedures within the data grid execute the standard I/O operations, enabling the same procedure to be used on different operating systems (Windows, Unix, Mac). The middleware then maps the standard I/O operations to the protocol required by the storage system. This makes it possible to support a wide range of clients when accessing data stored in multiple types of storage systems. The data grid acts as a broker mapping client protocols to storage protocols.

A second key aspect of the DFC architecture is that it enables federation. The systems where data are stored can be created independently of each other. The middleware that manages the logical name spaces runs as an application at each storage site. Data from the site are registered into the data grid. Policies are then applied within the middleware to control the properties of the shared collection. This makes it possible to interact with multiple types of community resources while preserving the autonomy of each site. The controls applied by the data grid are added on top of the controls applied within each community resource.

A loosely coupled federation architecture is shown in Figure 1. The architecture extends the ideas present within data grids to differentiate between research environments, collaboration environments, and community resources. A research environment may be supported on a laptop, or institutional cluster, or supercomputer. The researchers need to be able to acquire relevant data sets, import them into the research environment, conduct the analyses, and then save the results. The collaboration environment enables the researcher to share data and workflows, re-execute workflows, and track multiple versions of research results. All interactions with the collaboration environment are controlled by policies. This enables automated retrieval of data sets, as well as automated transformation of data formats. The community resources provide the information
catalogs and discipline-specific data repositories that contain the data that are being analyzed. The mechanisms used to access the community resources may be native protocols (file system I/O commands), web services, or brokers that translate between protocols. By using data grids, storage drivers can be applied that enable access to each of the loosely coupled federation components.

We use the DFC architecture to implement use cases that require the retrieval and analysis of data sets from multiple repositories. This is facilitated through development of repository specific drivers, or creation of soft links for registering data. Examples of repository specific drivers include a THREDDS\(^4\) server, for accessing variables within scientific data sets; the NetCDF\(^5\) and HDF5\(^6\) function libraries for data sub-setting and metadata extraction; a Windows file system; a Unix file system; a tape archive; and cloud storage. A soft link provides a basic function (micro-service) that can interact through the appropriate protocol with a remote repository. Examples of soft links include Z39.50, the Hydrologic Information System (HIS) from the Consortium of Universities for the Advancement of Hydrologic Science, Inc.\(^7\), ftp, and http. By choosing the appropriate storage driver or file soft link, data can be accessed across a wide variety of storage repositories.

The procedures that are executed within the data grid are composed by chaining together micro-services (Ward, 2011). The micro-services exchange information through in-memory structures, or through persistent state information maintained in a catalog, or through files, or through communication over a network. The in-memory structures can be serialized, sent over a network, and unpacked for use by a remote micro-service. This makes it possible to create workflows that are distributed across multiple storage locations and apply the micro-services where the files are located, and to build processes that are automatically applied every time a file is loaded into the logical collections. Within the iRODS data management system, more than 250 micro-services are provided that support basic operations that range from manipulation of files, to manipulation of metadata, to selection of storage resources, to exchange of messages, to invocation of remote procedures. The workflow language that manages the micro-services supports multiple data types (string, integer, binary, double, lists), arithmetic on variables, conditional tests, and loops. A query can be made to the metadata catalog, and the results can then be manipulated within a loop. A typical workflow would apply the following steps:

- Assign input parameters
- Query the metadata catalog to identify relevant input data sets
- Loop over the catalog results to set up input variables
- Acquire relevant data sets from external sources
- Project data set variables to a desired coordinate system
- Apply an analysis

---

\(^4\) Thematic Realtime Environmental Distributed Data Services, [http://www.unidata.ucar.edu/projects/THREDDS/](http://www.unidata.ucar.edu/projects/THREDDS/)

\(^5\) Network Common Data Format, [http://www.unidata.ucar.edu/software/netcdf/](http://www.unidata.ucar.edu/software/netcdf/)

\(^6\) HDF5, Hierarchical Data Format, [http://www.hdfgroup.org/HDF5/](http://www.hdfgroup.org/HDF5/)

• Store results back in the data grid.

By encapsulating operations within micro-services, the functions can be shared and re-used by other researchers. By chaining micro-services together, complex analyses can be assembled from re-usable components. By sharing the workflows, researchers can modify input parameters and track changes in the output results. Reproducible science can then be achieved, through sharing of not only the workflows, but also the input and output files. If the workflows execute iRODS micro-services, the workflows can be executed on any operating system for which appropriate I/O drivers have been written. This makes it possible to consider archiving workflows as well as data sets when preserving the results of a research initiative.

3. EcoHydrology Use Case
Researchers working in the interdisciplinary field of ecohydrology are concerned with the cycling of nutrients, energy, and water through coevolved climate-soil-vegetation systems (Rodriguez-Iturbe 2000), and with the interaction between water cycling and the ecological community. One modeling framework designed for research on the co-evolution of watershed hydrologic and ecological systems is the Regional HydroEcological Simulation System (RHESSys – Band et al, 1993; Tague and Band, 2004). This framework uses a general representation of watershed structure as a hierarchy of landforms, soils, and ecosystems connected by the coupled mass conservation of water, carbon, and nitrogen between land and atmosphere, and through surface and subsurface hydrologic flowpaths. Before running a RHESSys simulation, researchers must first process information on the topography, canopy cover, soils and hydrology of an area into a nested set of ecosystem patches, hillslopes, and catchments arranged through a connected drainage network, and encoded as a “world file” and a “flow table” (figure 2, bottom left). This nested watershed data preparation workflow incorporates terrain properties (e.g. slope, aspect, drainage direction) soil physical and chemical information, canopy physiologic properties and initial conditions (figure 2) built using information from: the U.S. Geological Survey (USGS; digital elevation model; DEM); NASA (canopy Normalized Difference Vegetation Index; NDVI) or other remote sensing information (aerial photography); U.S. Department of Agriculture (USDA; soils); and other, diverse data sources. Stream gauge, discharge and chemistry time series and meteorological information are accessed from USGS, National Climatic Data Center (NCDC) or other records (e.g. USDA Forest Service, Long Term Ecological Research (LTER) network sensors). The RHESSys data preparation workflow was originally developed as a set of largely manual downloads, reformatting, registration and additional spatial data processing (e.g. transformation of NDVI to leaf area index; LAI) using tools written in the GRASS GIS environment (GRASS Development Team 2012).
Additionally, post-processing workflows for analysis of output time series of variables of interest (e.g. soil moisture, litter carbon store) at the patch, hillslope or basin scale, or for fully updated world files representing all state variables of the system being simulated, are developed in a variety of tools ranging from GRASS, to R® statistical analysis software (R Development Core Team 2007) to desktop spreadsheet applications. Finally, output from RHESSys simulations can be used as input to related models (landslide, water supply, fire behavior).
Two primary input files to RHESSys—the world file (which describes the hierarchical representation of the landscape) and flow table (which describes the routes of lateral redistribution of water between patches in a hillslope)—are generated from the execution of a data preparation workflow (figure 2, bottom left). However, before the RHESSys simulation workflows can be executed, climate forcing data (e.g. minimum temperature, maximum temperature, and precipitation) must be supplied as input. Additionally, RHESSys model simulations are typically calibrated by comparing simulated streamflow to observed streamflow recorded by streamflow gages. Our current iRODS workflow leverages the development of CUAHSI Hydrologic Information System (HIS) to manage access to all point source time series climate and water data from USGS, NCDC and other sources (http://his.cuahsi.org/).

The nature of RHESSys model components linked for the co-evolution of transient ecosystems within the geomorphology and distributed hydrology of the watershed is to maintain a carbon balance supporting ecosystem survival, and growth for healthy ecosystems, based on the balance and adjustment of above and below ground biomass, and available water, nutrients and energy. Lack of balance can occur through unrealistic combinations of climate, soils, topography and standing biomass, which can lead to simulated ecosystem failure or significant adjustment and adaptation of the canopy so that it does not resemble key observed patterns (e.g. canopy LAI). This is unlike other models that will compute water and carbon cycling, but do not include growth and allocation and the potential for ecosystem failure (e.g. carbon starvation). The practical consideration of this constraint is that simple overlay and combination of natural resources data and remotely sensed canopy information typically needs to be augmented by specific decision points and intervention by modelers; local context and interdisciplinary knowledge are required. This can be seen in particular for modeling urban ecosystems, where national spatial data infrastructure are inadequate for modeling ecosystem processes that result from fine-scale spatial heterogeneity in land cover and management practices; high-resolution data (e.g. parcel, stormwater network, and aerial photography data maintained by municipalities and counties) and observations (e.g. household surveys or other sources of lawn fertilization or stormwater management practices gathered by researchers) are required.

Many modeling decision points are not fully documented and remain as tacit knowledge—passed from faculty to graduate student and among students—of the linked hydrologic-ecological systems based on field observations and experience. Therefore, ecohydrology modeling with RHESSys provides a case study requiring significant improvement in informatics tools to improve social interactions between disciplinary scientists, better codify research knowledge and experience, and provide for scientific reproducibility by managing and documenting the data-to-theory hierarchy including information provenance, access and workflows, and facilitate interoperability and integration of the separate disciplinary data and models represented in figures 2-3.

A prototype version of the RHESSys ecohydrology workflow presented in figures 2 and 3 has been implemented within the integrated Rule Oriented Data System (iRODS) data grid framework (Rajasekar et al. 2010). This workflow relies on the ability of iRODS data grids to enable collaborative research through the creation of virtual collections.
across institutional and disciplinary information sources that provide a unifying namespace for data. The elements of the shared collection can include soft links to data in other repositories and descriptive metadata to support search. In such a data grid framework, each research group within a community of collaborators can develop policies that control the properties of the shared data collection, including access constraints, required provenance information, and access services. The community-based policies are used to control procedures that automate administrative tasks, apply desired data format transformations, and validate collection properties. The procedures are cast as workflows that accurately capture provenance information (input parameters and files, processing steps, and outcomes). The information needed to re-compute an analysis are managed as an active object within iRODS, enabling the re-creation of scientific results when any of the underlying data sets are re-calibrated or otherwise changed. In the example of the RHESSys ecohydrology data preparation workflow that was implemented, datasets from multiple sources are represented as iRODS objects, with iRODS rules providing data transformation services (e.g. spatial and temporal subset, geographic coordinate system transformation). The iRODS objects consist either of static local copies of datasets that change infrequently (e.g. landcover or elevation data updated every 5+ years) or logical references to dynamic datasets accessed via web services interfaces (e.g. temperature and rainfall data, which are updated daily). In the case of logical data sources iRODS micro-services are used to fetch data as needed via web service-based interfaces, obviating the need to implement web services logic within the legacy RHESSys codebase.

Toward the end goal of supporting reproducible and collaborative ecohydrology modeling we have developed an automated iRODS-based RHESSys data preparation workflow that begins with: (1) determination of region of interest (ROI) based on a streamflow gage listed in the National Hydrography Dataset (NHD); (2) extraction of DEM tiles from the National Elevation Dataset (NED); (3) extraction of landcover tiles from the National Land Cover Dataset (NLCD). The DEM tile is imported into GRASS GIS, which is used to delineate watershed boundaries and develop the nested watershed structure (figure 2, center). Future versions of the workflow will include a similar iRODS-based approach to automate the incorporation of soil and vegetation properties (LAI, phenology). Given that NHD, NED, and NLCD data change infrequently, these data are made available to the RHESSys data preparation workflow by first manually downloading the data into an iRODS data grid. In future work we will incorporate access to other spatial data services accessible via web services interfaces, for example DEM data from DEM Explorer (http://ws.csiss.gmu.edu/DEMExplorer/), and soils data from SoilWeb (http://casoilresource.lawr.ucdavis.edu/soilsurvey/).

The advantages of the iRODS-based RHESSys data preparation workflow that was developed include: (1) integration across heterogeneous data models used by the different information sources in RHESSys; (2) implementation of the workflow through the development of specific, linked iRODS rules and micro-services; (3) documentation, archiving and access to the full data and information chain; and (4) potential to encode local site-specific context and knowledge developed through experience as part of conditional workflows. Currently, iRODS manages data access, including use of real-time calls to CUAHSI-HIS web services to identify and retrieve hydrologic point time
series information, data provenance, and simple workflows to build the RHESSys world files and flow tables including embedded GRASS GIS functionality, retrieval of forcing and calibration information, and execution of model runs. More complete workflows are under development with key implementation steps and research questions addressing the ability to create flexible workflows allowing user steering by encoding decision points currently used in manual workflows, and management of geoanalytics tools for higher-order visualization and analysis of model predictions.

4. Texas Drought Use Case
The 2010–2012 drought is a severe to extreme ongoing drought in the US South, including parts of Texas and other southern states. The worst effects have been in Texas, where near-record drought has parched the state since January 2011. Texas suffered an estimated $7.62 billion in crop and livestock losses. According to the National Climatic Data Center (NCDC), in 2010-11, Texas experienced its driest August–July (12-month) period on record. Drought analysis and long-term prediction is a major challenge in hydrology with both data-driven analysis and computationally-intensive modeling efforts involved. Researchers at the University of Texas recently organized a “Texas Drought 2012, are we prepared?” water forum to address the urgent needs in Texas with participants from multiple stakeholders in state and local government agencies and academic researchers. It is apparent that drought analysis and management is a multidisciplinary effort that requires data from multiple sources and collaboration across multiple domains. This is a classical geosciences scenario that requires immediate attention.

In collaboration with the researchers in Texas, we have picked an analysis area that has significant impact in studying the water supply and drought scenario in Texas. The area is located at the San Antonio and Guadalupe River Basins in Texas (see Figure 4). The computational modeling of drought analysis in this paper was based on the RAPID (Routing Application for Parallel computation of Discharge) river routing model (David et. al. 2011). Given surface and groundwater inflow to rivers, this model can compute flow and volume of water everywhere in river networks made out of many thousands of reaches. The design of RAPID allows it to be adapted to any river network, if given basic connectivity information. For the case study, multiple data input files are needed that include NHDPlus river network connectivity, lateral inflow from the land surface (computed with Noah-MP, a climate model) and gage measurements (from USGS NWIS) for a 4-year run (between 2004-01 and 2007-12) in the San Antonio and Guadalupe River Basins in Texas. The output file is a NetCDF file describing water flow rate in the river network and can be visualized in a GIS system such as ArcGIS.

In order to provide a reproducible and repeatable process for the drought analysis, a workflow is created using the Cyberintegrator workflow system to provide a step-by-step execution process management. Figure 5 shows a screenshot of the workflow in the Cyberintegrator editor. This workflow can be launched either in a desktop environment, a

---

9 http://www.jsg.utexas.edu/ciess/events/
10 http://www.geo.utexas.edu/noah_mp/default.htm
11 https://opensource.ncsa.illinois.edu/confluence/display/CBI/Cyberintegrator+Home
remote server, a public cloud or even a supercomputer. A web-based science gateway-style portal\textsuperscript{12} was created to allow the RAPID model to be executed on the NSF XSEDE supercomputing resources, since the RAPID model can be run in an MPI (Message Passing Interface)-supported parallel computing environment. Both the workflow approach and portal significantly reduce the barriers for re-executing and re-producing the analysis.

To provide further workflow reusability and cross-disciplinary collaboration, a workflow interoperability experiment was conducted between this RAPID workflow and the iRODS system. We leveraged the RESTful endpoints supported by the Cyberintegrator workflow and several features provided by iRODS including workflow registration, remote invocation of workflow execution and automatic downloading of workflow output into the iRODS repository. At the June 2012 EarthCube charrette meeting, we successfully demonstrated the interoperability between the Cyberintegrator workflow and iRODS using the RAPID model-based drought case study as an example. Such interoperability is critical for distributed collaborative workflow sharing and reuse.

Within the iRODS data grid, three micro-services were developed that provided the following functions:

1. Initiate a Cyberintegrator workflow
2. Monitor the progress and status of a Cyberintegrator workflow
3. Retrieve results from a completed Cyberintegrator workflow.

The first micro-service’s role is to launch a Cyberintegrator workflow using input data stored in the iRODS data grid. It retrieves an input object (e.g: CI-KICKOFF-PAYLOAD) from a given iRODS collection and uses it to make a POST request to \url{http://rapid.ncsa.illinois.edu:8183/cyberintegrator/engine/workflows/execute}. Upon success the HTTP response contains the ID of the newly started workflow, which the micro-service returns and saves as a workflow variable.

A second micro-service takes the previously returned workflow ID and makes a GET request to \url{http://rapid.ncsa.illinois.edu:8183/cyberintegrator/engine/workflows}. The response is in either XML or HTML and describes the status of the workflow (started, running, finished), which the micro-service parses. This step is repeated every 30 seconds until the workflow is found to be completed.

A third micro-service then takes the workflow ID and makes a GET request to \url{http://rapid.ncsa.illinois.edu:8080/earthcube/workflow/download} to retrieve the workflow’s output, made available for download by Cyberintegrator after the workflow is completed. The data are written to a new iRODS object in an iRODS collection, and is now part of the data grid.

This enables the automation of the management of both input and output files associated with each drought analysis. The above interactions between iRODS and Cyberintegrator are shown in Figure 6. The iRODS data grid controls the initiation and archiving of

\textsuperscript{12} \url{http://rapid.ncsa.illinois.edu:8080/rapid/}
results from a Cyberintegrator managed workflow. The Cyberintegrator system manages the execution of the workflow on HPC resources.

5. Hypoxia Use Case
Hypoxia refers to low levels of dissolved oxygen in the water. In the northern Gulf of Mexico, riverine and ground water hypoxia data have been collected for several decades for summer months since it is a large seasonal area of low dissolved oxygen. Different factors affect these seasonal oxygen levels including temperature, stratification, and currents of the coastal waters, causing an increase in the mortality rates of living organisms in the region.

Models for analyzing and predicting hypoxia size, location and strength are currently available, and require inputs for nutrient fluxes, stream discharges, ocean temperatures, etc. to execute. However, these inputs generally require access to data from several disciplines, e.g., terrestrial hydrology, physical and biological oceanography, from multiple data providers, e.g., National Oceanic and Atmospheric Administration\(^{13}\) (NOAA) and Consortium of Universities for the Advancement of Hydrologic Science\(^{14}\) (CUAHSI). In addition, the data from different disciplines and providers have different encoding formats, scales, and sampling geometries, which makes continuous integration of hypoxia data a challenge. Moreover, data retrieval from the multiple government agencies, requires implementation of different access protocols, and caching of data for subsequent analysis.

Often, when a hypoxia researcher needs access to data from multiple data providers, he/she needs to work with the technologies provided for access to their datasets. For example, hypoxia data collected by USGS and U.S. Environmental Protection Agency\(^{15}\) (EPA) are available via CUAHSI web services. CUAHSI web services are designed with a goal to make it easy to locate data collection sites within latitude-longitude bounding box and to get data for these sites within a specific time frame, making it easier to query the data on these key spatial and temporal aspects. However, the NOAA National Oceanographic Data Center \(^{16}\) (NODC) data is available as FTP downloads for datasets with multiple files, requiring merging the data files and cleaning them of bad values before loading the available data into a database to make querying easier. Scientific workflows can help with standardization of access to these different data resources while providing a unified interface to different technologies for access and integration of hypoxia data and, in turn, execution of hypoxia models that use the integrated data. In addition to the cross-domain challenges, such executable workflows need solutions that include a combination of workflow technologies, with services, layered architecture, brokering and standards-based approaches.

5.1 Scientific Workflows to Discover and Visualize Hypoxia Data

We have used the Kepler scientific workflow system\(^\text{17}\) (Ludaescher et al. 2006) to demonstrate usage of workflows to access and integrate a set of representative data sources. The Kepler scientific workflow system is developed by a cross-project collaboration to serve scientific researchers and developers. Since its initiation in 2003, a diverse set of projects encompassing multiple scientific disciplines have used Kepler to manage, process, and analyze scientific data. Inherited from Ptolemy II\(^\text{18}\), Kepler adopts the actor-oriented modeling (Goderis et al. 2009) paradigm for design and execution of scientific workflows. Kepler provides a graphical user interface (GUI) for designing workflows composed of a linked set of components, called Actors. Actors are the implementations of specific functions that need to be performed and communication between actors takes place via tokens that contain both data and messages. Actors execute under different Models of Computations (MoCs) implemented as Directors, which serve as workflow schedulers and execution engines in Kepler. The designed workflows can then be executed through the same user interface or in batch mode from other applications. In addition, Kepler also provides a provenance framework (Altintas et al. 2006) that keeps a record of chain of custody for data and process products within a workflow design and execution. This helps track the origin of scientific end products, enabling validation and re-execution of the experimental processes that were used to derive these scientific products.

For the hypoxia use case, scientific workflow scenarios that span across scientific domains and data sources helped us identify cross-domain issues including data and file formats, units, and temporal and spatial representations. The developed workflows had to match the differences in the data formats and solve challenges on data discoverability, accessibility and downloadability that are not standardized. Through implementation of demonstrations for these use cases, we further analyzed workflow implementation requirements related to integrating geoscience data from multiple providers. The integrated datasets are then saved as a part of a workflow in iRODS to be used as input for execution of hypothetical hypoxia models.

One requirement for integrating workflows is inclusion of the workflow environment as one of the layers within a higher-level collaboration environment. This requires defining the mechanisms for instantiating workflow systems within web-based or other collaboration environments to perform different steps in the end-to-end process for a scientific study, e.g., access to data repositories and coupled visualization services, the processing of the data by an analysis workflow created by the researcher, and the storing of the results back to the repository within the research environment. Figure 7 illustrates system components to make such an implementation possible. The Kepler workflow system retrieves and integrates data from multiple hypoxia data providers through existing web services and databases. The integrated data then gets saved in iRODS and can be submitted through Kepler for execution on various computational resources. The workflow system provides access to other workflow tools within Kepler and keeps track


\(^{18}\) Ptolemy II website: http://ptolemy.berkeley.edu/ptolemy-II/, 2012.
of the provenance of the data flowing through the system. The saved provenance information can then be saved into iRODS.

As a part of the implementation of the hypoxia workflow in Kepler, we used such an end-to-end scenario to illustrate the challenges of cross-domain interoperability. The developed workflows for each step of data integration, visualization and recording are executed and driven from a web portal. Users can access past plots along with the provenance traces of the workflows that generated these plots. These workflows use a special set of iRODS actors in Kepler along with standard Kepler actors to access web services and query databases. Various data processing tools in Kepler including the expression actor, and Matlab and R actors handle the data manipulation and transformations. In particular, our hypoxia demonstration:

a) allows users to retrieve and assess compatibility of data from different domains using CUAHSI data access services and FTP-downloadable NOAA NODC datasets (See Figure 8);
b) resolves and unifies the cross-domain data differences including units, temporal and spatial representations, and file formats;
c) annotates the datasets with fitness for use information and plots the user-selected data (using openly accessible Google charts) for cross-domain comparison on dissolved oxygen values;
d) saves the resulting plots in iRODS as an annotated collection so that the results can be reused next time users need to search for these resources (see Figure 9); and

e) stores the provenance of each execution in iRODS so that the saved plots can later be validated and/or reproduced.

As a part of the future work, we would like to save these workflows as first class objects in iRODS so that the workflow definition can also be stored and made searchable along with execution provenance allowing comparisons between multiple runs of the same workflow using different input datasets.

6. Management of the Scientific Workflow Environment

Scientific data collections are generated and assembled within a research context. Even though data are created by a particular scientist or group, the data have utility beyond the particular research context in which they are created. To increase this utility, more than the data are needed. The provision of information about the scientific computing environment in which the data are created (including its research context, systemic metadata, application provenance and descriptive metadata) increases the sharability and interoperability of scientific data. One can categorize three different aspects of sharability and interoperability:

- **Re-test** (or reproduce) to check if the data are indeed correct. This is needed to evaluate, verify and confirm the theories and results of the research. The scientific context will be the same, but the application or experimental environment might be different. An example is where data are gathered to test and validate a
hypothesis in a controlled scientific environment and then repeating the same analysis in a different system or location.

- **Re-use** to apply the data in another, but closely related context. This is needed to use the data in subsequent analysis in a new research experiment as input for furthering research in the same domain. An example would be to use data generated by an earthquake simulation, to be applied to an engineering model in designing buildings.

- **Re-purpose** to apply the data in an entirely different context. This is needed so that data generated and verified in one context might be useful in another context in possibly an entirely different domain. An example is use of climate change data to predict the impact of drought on the ecology, and drive the development of drought-resistant plants.

One can view the three kinds of applications as a fundamental basis for sharing data. One can see that the three application types need the capture and association of increasingly stringent context information in order to be effective.

### 6.1 Scientific Workflow Environment

A scientific workflow environment provides a milieu for conducting data-intensive science. It includes access to large data collections, an execution environment for applying large and possibly distributed scientific workflows, the computing, networking and storage facilities for performing the data intensive analysis, and finally tools and services (middleware and applications) for enabling scientists to use the environment effectively for conducting research and gathering data.

As shown in Figure 10, creators of a scientific data set are typically driven by a research question (*purpose*), and choose domain information (*properties*) that they need to generate or collect in order to explore and validate their research question. One can view these properties as data and metadata that are collected or generated during the scientific process including guidelines, procedures and boundary conditions for the experiment. To generate the data collection, methods and applications (*procedures*) are invoked.

Once a data collection is created, the scientists can use it to test their hypothesis. But to make it a truly sharable and interoperable collection that can be used by other scientists and possibly used across domains, additional information is needed. One important aspect is the need to make sure that the properties conform to standards to make them useful to others. Such desired baseline conformance may include specification of the data format, and descriptive semantics, or constraints on the entire collection such as completeness, authenticity and usage restrictions. One can view these as additional properties on properties (meta-properties). To ensure that the conformance is true, enforcement of each meta-property can be controlled through a controlling authority (*policy*) that defines when, where and how the required meta-properties are achieved. Conformance of these meta-properties can be recorded in a database (*persistent state*) that is maintained alongside the main data collection. The verification of these meta-properties can be done by implementing assessment criteria as meta-policies that periodically query the collection, and re-apply procedures to verify that properties have been maintained (property verification). Rules may also repair problems to enforce
properties. Figure 10 defines the layered interaction of these elements of a digital scientific environment.

Properties provide the context that governs the creation of the data within the original research. But to be effectively sharable, one needs to gather additional properties. One can define a set of criteria about what properties are needed to enable effective data sharing among scientists, in the same domain and across domains. The criteria may include properties that each data set must possess (such as physical units), or properties that are related to the entire collection (such as completeness and coverage). The properties can be turned into assertions that the data set creators make about their collection. An outcome of the existence of these properties is improved sharability and interoperability by:

1. Encouraging consensus in data interchange among data providers and between data providers and data consumers
2. Provisioning of machine-oriented mechanisms for data access, management and interoperability with a desire for publishing in human accessible form the said mechanisms,
3. Enabling encapsulation of services under the “data as a service” regimen using machine-oriented and human-oriented data exchange formalisms.

6.2 Workflow Registration and Sharing

Workflows form an important part of the Scientific Research Environment. The iRODS system provides facilities for accessing internally stored (data grid stored) data as well as externally stored data. It also provides facilities for performing server-side operations (micro-services) and for chaining them into complex workflows. As described in Section 2, the workflow language provides sophisticated flow control like loops and iterations, conditional branching, recursion, a distributed and robust parameter passing environment, message passing, and context management environment. In version 3.0 of iRODS, support was provided for execution of server-side workflows and for access to data in non-iRODS repositories. However the management of workflows and its products was missing. The ability to share workflows was very much needed to enable collaborative research.

A major goal of any scientific endeavor is the demonstration of reproducible science. A simple interpretation of this requirement is the ability to share, inter-operate and re-execute analysis workflows in different ways. The approach taken by the iRODS development project is to use the same mechanisms to interact with workflows as are available to access data sets.

This introduces a new concept of realizable (iRODS) objects. A realized object is an extension to the file system that associates an “executable” process with an iRODS logical file name. The executable can be an invocation of a micro-service or the execution of a workflow. The associated iRODS logical object holds information about all of the environmental and parametric information needed for effective execution of the realized object. When the iRODS logical file is accessed, the executable is invoked. The results of the executed process are considered to be the file object that is returned from the access invocation. This just-in-time realization is quite different from accessing a
static file stored in a file system. The resulting file is stored as a separate (but associated file) in iRODS that can be used in future access. This approach makes it possible to store provenance information within iRODS, invoke the provenance information through any iRODS client, and instantiate a copy of the result as a file within iRODS. In our workflow-based realized object, the workflow represents the provenance information, including parameters for the workflow process. The execution of the workflow generates an output file that is registered into iRODS as a replica of the realizable object. Note that the workflow is implemented by chaining together iRODS micro-services, which can invoke processing locally within the iRODS data grid, or invoke the execution of an external process, or invoke a remote web service, or invoke an external workflow system. If the workflow uses as input files other realizable objects, it is possible to build a processing pipeline. Each realizable object in turn invokes the instantiation of its input files. The iRODS data grid can logically execute the entire chain of realizable objects, and by checking modification times determine whether the bottom-most object in the chain has been changed. Each realizable object in the chain can then be rebuilt if needed to ensure that all derived data products represent a consistent state. If none of the realizable objects within the chain have been modified, then the instantiated copy can be returned directly to the user.

The workflow-based realized object is a generalization of the concept of mime types, with the definition of an application mime type associated with workflows. Each application mime type has an associated directory that contains the products generated by execution of the application. For workflows, the associated directory contains the input files, a “run” file, an output directory, and the output files. This is shown in Figure 11.

The workflow text file, eCWkflow.mss, is ingested into an iRODS data grid. This file contains the text associated with the execution of an iRODS workflow. A structured directory, eCWkflow, is created through the iRODS mounted collection capability. This is a directory that is directly linked to the workflow file. The directory cannot be associated with any other workflow file. When a user puts an input parameter file into the eCWkflow directory, the iRODS data grid automatically generates an associated “run” file, eCWkflow.run. When the “run” file is accessed with an iget command, the system automatically uses the associated input file, and executes the associated workflow. An output directory is automatically created, eCWkflow.runDir0, and all output files are written to the directory. If a new input file is put into the workflow directory, another associated “run” file is automatically generated. Execution with the new input file forces the creation of a new output directory. This approach tightly couples input files and output files to a workflow. If the workflow is changed, a new workflow directory is needed to hold the associated input files. Re-execution of a workflow causes a new version of the output directory to be created.

This approach enables reproducible science. For each workflow, the input and resulting output files are tightly linked. Results from the use of different input files can be compared. A collaborator will be able to re-execute a workflow, modify input parameters, and compare output results.

This approach also enables collaborative research. A researcher can share a workflow, share input files, and share output files. The collaborator can associate a new mounted
collection with the workflow, and manage his or her input and output files separately from the original researcher.

The workflow can invoke external executables, including traditional workflow systems such as Kepler and Pegasus. If the workflow is constructed from iRODS micro-services, the workflow will be executable across all of the operating systems to which the iRODS data grid has been ported. This enables the re-execution of workflows in the future, ensuring that a scientific result that was generated today can be re-generated at any point in the future.

7. Conclusion
Data driven science research projects manipulate data records that have been assembled and stored in multiple federal repositories. Groups that collaborate on the analyses need to be able to share input files, share analysis workflows, and share results. Collaboration environments provide these capabilities. The same mechanisms are needed for reproducible science. In this case, the collaborator may be a researcher in the future who wants to verify a research result. The demonstration that collaboration environments can also enable reproducible science is based on the execution of geoscience use cases. Workflows that have been constructed for analyzing ecohydrology research questions, drought, and hypoxia can be implemented within the collaboration environment, shared, and preserved for future re-execution. The use cases that are discussed in this paper were presented at the NSF EarthCube charrette held in Washington DC on June 12-14, 2012.

8. References


Table 1. Logical Name Spaces Used in Federated Architectures

<table>
<thead>
<tr>
<th>Name Space</th>
<th>Operations</th>
<th>Virtualization interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Users</td>
<td>Authentication, authorization, groups</td>
<td>GSSAPI / PAM</td>
</tr>
<tr>
<td>Objects</td>
<td>Partial I/O, move, copy, replicate, share</td>
<td>Posix I/O &amp; staging</td>
</tr>
<tr>
<td>Collections</td>
<td>Organization, Browsing</td>
<td>System metadata</td>
</tr>
<tr>
<td>State information</td>
<td>Add, update, delete, query</td>
<td>Catalog interface to DBMS</td>
</tr>
<tr>
<td>Resources</td>
<td>Load leveling, fault tolerance, grouping</td>
<td>Storage drivers</td>
</tr>
<tr>
<td>Policies</td>
<td>Management, administrative, verification</td>
<td>Policy language</td>
</tr>
<tr>
<td>Procedures</td>
<td>Basic functions on each name space</td>
<td>Workflows</td>
</tr>
</tbody>
</table>
Figure 1. Loosely Coupled Federation Architecture

Figure 2. RHESSys workflow to develop nested watershed parameter file (worldfile) containing nested ecogeomorphic object framework, and full (initial) system state.
Figure 3. RHESSys workflow (cont’d) with initial worldfile ($T_0$), climate forcing data (LHS), observed stream discharge and chemistry (from CUAHSI HIS get calls), output of updated full system state (worldfile) and water, carbon, nutrient storage and flux time series and links to additional environmental models.
Figure 4. Drought Use Case Study Site: San Antonio and Guadalupe River Basins in Texas

Figure 5. RAPID drought analysis workflow in the Cyberintegrator workflow editor environment.
Figure 6. Integration of Cyberintegrator with the iRODS collaboration environment.

Figure 7. Using Kepler for hypoxia data integration and model execution while saving the results and provenance on iRODS.
Figure 8. Workflow for retrieval of data from CUAHSI and NODC.

Figure 9. Saving the plots of the integrated datasets in iRODS.

Figure 10. Concepts controlling data management applications
Figure 11: Automation of the Linking of Workflow Input and Output Files