Functional Requirements for Research Data Repositories*

Suntae Kim**

ARTICLE INFO

Article history:
Received 12 February 2018
Revised 20 February 2018
Accepted 27 February 2018

Keywords:
Repository,
IDR,
Institutional Data Repository,
Data Repository,
Functional Requirement

ABSTRACT

Research data must be testable. Science is all about verification and testing. To make data testable, tools used to produce, collect, and examine data during the research must be available. Quite often, however, these data become inaccessible once the work is over and the results being published. Hence, information and the related context must be provided on how research data are preserved and how they can be reproduced. Open Science is the international movement for making scientific research data properly accessible for research community. One of its major goals is building data repositories to foster wide dissemination of open data. The objectives of this research are to examine the features of research data, common repository platforms, and community requests for the purpose of designing functional requirements for research data repositories. To analyze the features of the research data, we use data curation profiles available from the Data Curation Center of the Purdue University, USA. For common repository platforms we examine Fedora Commons, iRODS, DataONE, Dataverse, Open Science Data Cloud (OSDC), and Figshare. We also analyze the requests from research community. To design a technical solution that would meet public needs for data accessibility and sharing, we take the requirements of RDA Repository Interest Group and the requests for the DataNest Community Platform developed by the Korea Institute of Science and Technology Information (KISTI). As a result, we particularize 75 requirement items grouped into 13 categories (metadata; identifiers; authentication and permission management; data access, policy support; publication; submission/ingest/management, data configuration, location; integration, preservation and sustainability, user interface; data and product quality). We hope that functional requirements set down in this study will be of help to organizations that consider deploying or designing data repositories.

1. Research Objectives

Research data must be testable. Wikipedia highlights that: “Science is a systematic enterprise

* This research was supported by Korea Institute of Science and Technology Information (KISTI).
** Researcher, Korea Institute of Science and Technology Information, Korea (stkim@kisti.re.kr)
International Journal of Knowledge Content Development & Technology, 8(1): 25-36, 2018.
http://dx.doi.org/10.5865/IJKCT.2018.8.1.025

that builds and organizes knowledge in the form of testable explanations and predictions about the universe". To make data testable, tools used to produce, collect, and examine data during the research must be available. Quite often, however, these data become inaccessible once the work is over and the results being published. As Brown et al. (2017) points out, research results can show inconsistency depending on the nature of the supporting data. Accordingly, the problem of preserving and reusing data within the relevant context becomes increasingly important. Open Science is the international movement that addresses this problem.

The core of the Open Science movement are Open Access and Open Data initiatives aimed at overcoming legal, organizational, and technical barriers in accessing research findings and data. Open Science is actively promoting the concept of data, which lies at the heart of the fourth industrial revolution, with the focus on data preservation and reusability. These problems are also receiving an increasing attention on the part of research community and general public. Especially, international community is showing unanimity on that the data obtained as a result of projects supported by public funds must be made available. Thus, on the recent Ministerial Meeting of the OECD Committee for Scientific and Technological Policy, Open Data became the central topic, which further demonstrates the significance of the trend.

Accordingly, the purpose of this study is to determine functional requirements for research data repositories so that scientific data obtained as a result of projects supported by public funds could be properly preserved, managed, and reused.

2. Research methods

In order to derive functional requirements for research data repositories, we examine the features of the research data, common repository platforms, and community requests.

For the features of the research data, we use Data Curation Profiles available from the Data Curation Center of the Purdue University, USA. Data Curation Profile provides detailed information on particular data forms that might be curated by an academic library (Witt et al., 2009). For common repository platforms, we examine Fedora Commons, iRODS, DataONE, Dataverse, Open Science Data Cloud (OSDC), and Figshare. These platforms are used a lot as a data repository tools globally. We also analyze the requests from research communities such as RDA Repository Interest Group and DataNest users. To design a technical solution that would meet public needs for data accessibility and sharing, we take the requirements of RDA Repository Interest Group and the requests for the DataNest Community Platform developed by the Korea Institute of Science and Technology Information (KISTI).

We do not consider system requirements, however, but restrict our analysis to mapping functional requirements of the RDA Repository Interest Group.

3. Previous research

Gibbons (2004) demonstrated how data stewardship, efficiency, showcases, wider distribution, and scholarly communication affect the effectiveness of repository management. He also points out that there is a pressing need to promote research of Institutional Repositories (IR) that recently started to evolve into Institutional Data Repository (IDR) and draw great academic attention. Below we list some recent research on the subject.

In the era of Big Data, there is an urgent need to handle in real time extremely large volumes of data from IoT devices. When it comes to real-time handling of such amount of heterogeneous data, however, the former relational databases show several technical constrains. Accordingly, NoSQL technology for data repositories is receiving increasingly more attention. Kang et al. (2016) came up with a MongoDB design for RFID and IoT device data. Harvey et al. (2017) designed data repository by applying FAIR Data principles and using DataCite schema-based metadata and ORCiD. Boukhari et al. (2018) suggested a physical schema for supporting repositories, such as database or data warehouse, throughout the entire life-cycle. Many researches study data consolidation, data acquisition, data transfer, data uploading and other related issues. Considerable effort is also put out in maintaining domain-specific vocabularies to ensure terminological consistency. Given the importance of data reproducibility in healthcare, Johnson et al. (2018) designed MIMIC Code Repository, a Knowledge Base for clinical evidence and medical conclusions. This repository helps reproduce research findings and make medical decisions based on detailed information about the severity of the disease, associated illnesses, sepsis-associated conditions and physiological cycles, multiple organ dis-function, and other critical care data.

As well as being very important for general public, data are also instrumental for business. A great deal of effort is going into systematic management of the enterprise data and enhancement of integrated data services for decision making. The joint data repository project between UNC's Kenan Institute of Private Enterprise and Duke Innovation and Entrepreneurship Initiative (I&E) of the Duke University is a good example of how these problems are addressed. Recently, we also observe the increasing number of patents related to repository management. Especially, patents related to security management of repository entities. Keyes et al. (2017) conducted patent examination and filed a patent for entity-based asset security for repository system.

Thus, we can see that previous studies focus mostly on data storages (Kang et al., 2016), meta-data-based repository design (Harvey et al., 2017), data repository lifetime management (Boukhari et al., 2018), domain-specific terminology management (Johnson et. al. 2018), entity security in repository system (Keyes et al., 2017), etc. So far, however, there has been little study on functional requirements for data repositories. In other words, no research deals with requirements that could be generally applied to the design of research-oriented repositories. This study therefore set out for practical reference on design of the repository itself.

4. Results and Discussion

4.1 Research data features

We examined 13 data features from Data Curation Profiles (DCP) available from the Data Curation Center of the Purdue University, USA, for research data in biotechnology, astronomy and space science, geology, chemistry, humanities, etc. Analysis results and the size of data greatly varies depending on subject field. There are many factors that affect the choice of the platform, such as support for metadata and distributed storage, whether the files contain structured or unstructured data, images, audio, video or other formats. At the same time, this is the most important feature when a platform is considered for storage and retrieval of heterogeneous research data.

As for data types, there are numbers, sequences, graphs, multimedia, and other types. In terms of data editing, there can be a need for keeping record history, deduplication, compression, etc. Also, speaking of data relationships, there can be a need for logical and physical organization of data storage and retrieval, as well as load balancing. In terms of data retention, the capacity of the storage system and the capability of automatic archiving become increasingly important.

It has been found that among the available profiles for the genom data, data on mutual influence of proteins, astronomy and space data, the genom sequence data set contains the biggest size of unstructured data that come in series. These data is further editable and relationships among the data are preserved. The data on mutual influence of proteins is of relatively small size and contains mostly structured graphs. Like for the genom data set, these data can be edited and relationships among the data are preserved. On the other hand, astronomy and space data greatly varies in size and come mostly in image and multimedia formats. Unlike for the genom and protein data sets, these data cannot be edited.

4.2 Research data repositories

In this section we examine common research data repository systems and come up with some system requirements. For the purpose of this study we analyze Fedora Commons, iRODS, DataONE, Dataverse, Open Science Data Cloud (OSDC), and Figshare.

4.2.1 Fedora Commons & iRODS

Fedora Commons is a powerful modular open source system for management and provisioning of digital content. It can also be used for data storage and acquisition. For data management REST and SOAP Web service interfaces are provided. Architecturally, main data are stored in XML files. The system maintains RDF indexes, and depending on the content model makes it possible to attach different services, including dynamic ones. With the help of plugins, users can further apply different security policies to their content. There are no constrains on data types and formats, and extra metadata are also supported. Users can create, edit, and delete metadata that comply with W3C Linked Data Platform (LDP) 1.0. Also, distributed in-memory key/value storage architecture

is supported, while Apache Server provides load balancing with mod_jk clustering that further enables running concurrent multiple workers.

iRODS is an open source data management repository system. It supports virtualization, data discovery, automated workflows, collaborative security, etc. For research data management, metadata and distributed storage based on IES (iCAT-Enabled Server) resource servers are supported. For live data concurrent file and operating system are used. All loads (CPU, memory, Run, Swap, Page IO, Network, Disk usage, etc.) can be balanced by configuring the allowed thresholds. For database, users can select among PostgreSQL, MySQL, and Oracle. Command mode for data transfer is also supported. Different operating and file systems are concurrently supported, which makes it possible to select configuration that suits the best the specific character of the research data.

4.2.2 DataONE and Dataverse

DataONE is a system for storage and management of metadata created in different repository systems. Data can be uploaded in various XML formats. Member nodes provide physical data storage and management of metadata relationships so that information across different repositories can be retrieved. To search metadata across physical nodes (three nodes in the configuration at hand), Solr engine is used to run distributed queries. Special analysis, visualization, and metadata management tools are further provided for research data on environmental science. For research data transfer RESTful API is supported.

Dataverse in turn provides a service for storing geospatial images and astronomy data in a compressed format. Backend Solr engine is used to distribute search query load with front-end Apache working as a web server. Live data are stored on SAN (Storage Area Network) disks, while an RDBMS is used for metadata relationships management. Reserve server provides support for R and other application languages, which can be used concurrently. The search engine and the metadata management server can be in active or standby mode. As a database PostgreSQL is used. Network load is balanced on the physical layer.

4.2.3 Open Science Data Cloud (OSDC) and Figshare

OSDC offers project-oriented cloud repository services. Depending on project data type, distributed storage and management are further available. With the help of OpenStack technology, individual servers running virtual machines with preinstalled components are provide as a cloud service. A wide variety of data types, including formats for biology, genomics, earth science, social science, text data, astronomy, music, model reduction, etc. are supported. Users can decide how they want to manage distributed storage by selecting OpenStack object storage, Hadoop or other technology. They can also install the required additional software packages on the virtual machines and access open data by identifiers.

Figshare in turn is a repository service intended for academic paper management. It is based on Amazon S3 services, which support distributed storage management and load balancing. Special APIs for data uploading and downloading are further offered. Users can save all data associated

with their paper. Amazon S3 writes these data to distributed disk files, hashes them, and stores internally. Load balancing is achieved through automatic data replication and recovery, scaling up, and additional cluster nodes.

The examination above shows that data storage and management, data search and retrieval, data acquisition and transfer make up the core functionality of any research data repository. As far as distributed storage is concerned, it is important that large data, depending on their nature, could be partitioned and stored across distributed storage devices. Small-size data should be also accounted for particulars, and data kept together on a specific storage device should be categorizable. In terms of load balancing, it should be possible to distribute the workload across different nodes. Also, it should be possible to distribute data across the nodes to avoid capacity problems in case of large amounts of data. It is especially important that the system could replicate jointly analyzed data across the different nodes. Also, data archiving is quite important for load management of large-size data. As for data storage and management, authentication and permission management assume a great significance. With data search and retrieval, the key functionality includes the speed of data updates, data change versioning, and the ability to manage extended metadata. Physical and system scale-out is required, while library extensibility should be further supported. In order to support diversified search interfaces, the corresponding APIs should be provided. Speaking of data collection and transfer, Dataserve and DataONE both support data transferring with simultaneous compression.

4.3 Community requirements

This section deals with community requests on different functional requirements for research data repositories. We analyze these requirements based on discussion held in RDA (Research Data Alliance), which is a consultative body for research data, and DataNest Community Platform developed by the Korea Institute of Science and Technology Information (KISTI).

4.3.1 RDA Repository Interest Group

RDA aims at overcoming legal, organizational, and technical barriers in seeking data openness and sharing. RDA Repository Platforms for Research Data Interest Group is a special group to discuss functional requirements for repository platforms. Although on March, 2017, within the framework of the 7th RDA Plenary meeting, the group session noticed that at the time functional requirements had been lacking categorization consistency, by the time of this study this became more systematic. Functional requirements for research data repositories can be largely grouped into 13 categories (metadata; identifiers; authentication and permission management; data access, policy support; publication; submission/ingest/management, data configuration, location; integration, preservation and sustainability, user interface; data and product quality).

See <Table 1> for details of the categories proposed by the RDA Interest Group.

Table 1. Functional Requirements for Research Data Repository Platforms by RDA IG

Category	Functional requirements
Metadata	Support for different metadata schemas (+3)
Identifiers	PID assignment for data management (+1)
Authentication and authorization	Provide integration with external systems (+1)
Data access	Choose levels of data access (+6)
Policy support	Allow use of policies for data processing, quality management, etc. (+2)
Publication	Provide data access statistics (+3)
Submission / Ingestion / Management	Support remote access and management of the repository (+12)
Data organization	Allow collection virtualization and logical naming
Location	Support for high performance computing and tight integration with data processing
Integration	Support interoperation among individual data management systems (+1)
Preservation and sustainability	Maintain a permanent history of versions for all data (+3)
User interface	Individual and organization-wide collections (+1)
Data and product quality	Capture “degree of confidence” on each data item

4.3.2 DataNest Community

DataNest is being developed by Korea Institute of Science and Technology Information since 2012 as a research data repository for Korean institutions. The project is based on DSpace and Fedora Commons. DataNest is developed based on community feedback on what is needed to enhance existing features and create new functionality. The discussion below outlines DataNest community requests on functional requirements.

In metadata management, it is requested that users could add custom fields to the existing collection schema. In interworking, automatic repository registration with OpenDOAR, ROAR, re3data.org, Databib is requested. Also, data submission should be subject to approval by email, interworking among repositories is required. In terms of product quality, the system should offer necessary security features. Versioning should be supported for data submission, ingestion, and management. Personal data should be treated in a secure manner. Also, in case of co-authorship, it is necessary that all authors could check and verify the data or representative checkup mechanism should be provided. In addition to administrative rights, there should be per collection rights and support for uploading and re-using templates.

User interface requests (10 in all) are the most common. There is a need for test area before actual data are submitted. In addition to keyword search and data browsing, it should be possible

to do custom search inside specific collections. Search queries should support syntax highlighting. Unauthorized users should not be able to download data. Raw data should be made available only to specific users. There must be a way to extract SNS data. Users must be able to configure the system based on provided profiles. When data is submitted, users must be able to attach files to the required fields. Search interface should support search by organization or collection name and specialized functions for working with numeric data. Lastly, it should be possible to split and merge collections.

5. Functional requirements for research data repository

This section outlines the functional requirements for research data repository. Having examined the functional requirements from data repository platforms, community requirements from the RDA and DataNest communities and the features of the research data by using the data curation profiles, we determined the following 13 categories and 75 functional requirements. We do not consider system requirements, however, such as distributed in-memory key/value data storage, load balancing through load distribution management, and the like. It passes the bounds of this article's scope. I follow the category of the RDA and merged the requirements under it. To make the requirements merging process easier, some concepts were being made into the broader concept and some concepts were being made into the narrower concept in more detail. The examples of the former case are changing the 'Persistent identifiers' into 'Identifiers' and 'User Experience / User Interface' into 'User Interface'. The examples of the latter case are changing the 'Authentication' into 'Authentication and authorization' and 'Integration' into 'Integration / Interworking'. The each item of the functional requirements are merged or divided in more detail. So It is meaningless to identify the source where the requirements come from.

Table 2. Functional Requirements for Research Data Repository Platforms

Category	Functional requirements
Metadata	Support for different metadata schemas, support for domain-specific metadata, allow extended metadata for interoperability
	Support data labeling by data owner, authorized persons or automatic metadata extraction tool
	Provide metrics for metadata quality assessment
	Support storing XML files
	Provide metadata management tools
	Support search engine indexing
	Allow adding fields to collection schema
Identifiers	Support assignments of PID and DOI
	PID assignment for data management
Authentication and authorization	Support integration with external systems
	Provide single sign-on or support multiple authentication methods (Shibboleth, LDAP)

Category	Functional requirements
Data access	Support data access levels
	Support multiple interfaces such as WebDAV, FUSE, Java I/O, Python, Shell commands, REST and SOAP (F)
	Support multiple data versions
	Allow restriction of data useful time (by date and period)
	Manual and automatic search
	Support convenient submission and integration with external publishers
	Support data download; provide download API
	Support command mode for data transfer
	Provide RDF-based index
Policy support	Allow use of data policies (data processing, quality management, etc.)
	Support administrative rules for using policies
	Support data properties (licensing, security info, etc.)
Publication	Provide data access statistics
	Provide information for bibliographic citation, allow the export of bibliographic data to citation software
	Maintain citations linked to the data
Submission Ingestion Management	Allow automated task running (data acquisition, integration with analyzing tools, interaction with external software by related groups)
	Maintain a record of information flow into and out of the repository and provide the event log and current status
	Provide tools for managing data completeness and product quality (bit preservation, replication, checksum of data, metadata completeness, accuracy, correctness)
	Support workflow chains with microservices Allow dynamic service hook-up
	Support I/O streams for fast data transfer (ingestion, extraction)
	Provide mobile device support for laboratory notebooks
	Support remote access and management of the repository
	Provide an API to support workflows and uploads for data submission
	Allow managing heterogeneous data across multiple files
	Allow product developers to update product information
	Allow content to be marked for deletion by authorized users
	Provide vocabulary service (governance) for data reuse
	Provide information on data status (submission states: raw, processed, curated, published)
	Provide download history management (ideally, allow submitter to verify the data for secure data sharing)
	Provide technical means for secure collection of personal data
	Allow one person in the group to be assigned rights for representative data verification
	Allow one person in the group to be assigned rights for representative verification
	In addition to system administrator, allow administrators to be assigned for specific organizations and collections
	Allow data submission based on the previously uploaded templates

Category	Functional requirements
Data organization	Allow collection virtualization and logical naming
	Allow to split and merge collections
Location	Support for high performance computing and tight integration with near data processing
Integration / Interworking	Support interoperation among individual data management systems (Federation)
	Support store drivers (mapping access and storage protocol)
	Provide support for different file and operating systems
	Provide support for R and other application languages
	Provide support for analysis and visualization tools
	Support automatic registration with OpenDOAR, ROAR, re3data.org, Databib
	Provide email support for data submission confirmation and verification requests
Preservation and sustainability	Maintain a permanent history of versions for all data
	Support accessible data conversion formats
	Support extendable storage (scale-up, adding nodes to cluster)
	Register workflows as executable objects
	Support distributed data storage
User interface	Provide seamless and consistent interface
	Support individual and organization-wide collections
	Allow configuring collections by project or other units
	Provide a menu where users can access a test area to test their data before submission
	Allow custom search inside specific collections Provide dynamic search interface for specific collections
	Provide highlighting for search queries
	Allow download restrictions, block unlimited downloads (allow clicking by file name)
	Allow enabling/disabling all internal and external data services; allow specific users (project administrator, etc.) download raw data
	Allow extracting data in different formats, provide SNS integration
	Allow users specify subject profile
	Allow file uploading for each metadata field, when needed
	Allow search by community, organization or collection for data aggregation
	Allow numeric data to be viewed in table or chart format
Data and product quality	Capture "degree of confidence" on each data item
	Provide information on system security features

6. Conclusion

The objectives of this research are to examine the features of research data, common repository platforms, and community requests for the purpose of designing functional requirements for research data repositories, except for system requirements. As a result, we particularize 75 requirement items

grouped into 13 categories.

For metadata management, we suggest 7 items, including support for metadata schema. For identifiers, we suggest 2 items, including PID assignment for data management. For authentication and authorization, we suggest 2 items with regard to external integration, and 9 items with regard to data access and access levels. For policies, we suggest 3 items related to publication. Most requirements fall into data submission, ingestion, and management category, which is the core functionality of a data repository. All together this category has 19 items, including remote access and management. One item addresses physical location, logical naming, and virtualization of collection where we suggest high performance computing and tight integration with near data processing. Also, data submission, ingestion, and management is substantially about subsequent data retrieval so that research data could be used to verify the findings or examine the data in a different manner. Here, we suggest 6 items around integration and interworking, including analysis and visualization tools. For preservation and sustainability we suggest 5 items, including full data versioning. Also, many suggestions around user interface are related in fact to data submission, ingestion, and management. Here we suggest 13 items, including support for individual and organizational collections. Finally, with regard to data and product quality we suggest 2 items, including the degree of confidence for data items, and overall system security.

Data repository design is a multistage process, and managing functional requirements is the most time-consuming task. Especially, it is of paramount importance to examine existing solutions and set down precise requirements. Accordingly, hands-on results of this study can be used for practical implementation of functional requirements for research data repository. However, we deliberately do not cover system requirements in this study. Designing system requirements for research data repository would take an in-depth examination of cloud environment, networking technology, and available hardware and software. For a real-life research data repository design this information should be considered together with functional requirements suggested in this study. Further work needs to be done to establish system requirements.

As Gibbons (2004) points out in the data-intensive era research data repositories should keep not only research papers, but also all supporting data, while IR shifts the burden and responsibility of stewardship from the individual's level to the institution's level. This means that the demand for data repositories will continue to grow. And we hope that functional requirements set down in this study will be of help to organizations that consider deploying or designing data repositories.

References

Boukhari, I., Jean, S., Ait-Sadoune, I., & Bellatreche, L. (2018). The role of user requirements in data repository design. *International journal on software tools for technology transfer, 20*(1), 19-34.

Brown, G. W., Ouimet, P. P., Robinson, D. T., & Zoller, T. (2017). Understanding Entrepreneurship: Facilitating Academic Research with a Shared Data Repository.

DataONE. (n.d). Retrieved from https://www.dataone.org/

Dataverese. (n.d). Retrieved from https://dataverse.org/

Fedora Commons. (n.d). Retrieved from http://fedorarepository.org/about

Figshare. (n.d). Retrieved from https://figshare.com/

Gibbons, S. (2009). Benefits of an institutional repository. *Library Technology Reports, 40*(4), 11-16.

Harvey, M. J., McLean, A., & Rzepa, H. S. (2017). A metadata-driven approach to data repository design. *Journal of Cheminformatics, 9*(1), 4.

iRODS. (n.d). Retrieved from https://irods.org/

Johnson, A. E., Stone, D. J., Celi, L. A., & Pollard, T. J. (2017). The MIMIC Code Repository: enabling reproducibility in critical care research. *Journal of the American Medical Informatics Association, 25*(1), 32-39.

Keyes, D., Palanisamy, M., DiFranco, D. E., & Chin, D. M. (2017). *U.S. Patent No. 9,779,261*. Washington, DC: U.S. Patent and Trademark Office.

OpenScienceDataCloud (OSDC). (n.d). Retrieved from https://www.opensciencedatacloud.org/

RDA. (n.d). Retrieved from https://rd-alliance.org/about-rda

RDA: Functional Requirements for Research Data Repository Platforms. (n.d). Retrieved from https://my.usgs.gov/confluence/display/cdi/RDA%3A+Functional+Requirements+for+Research+Data+Repository+Platforms

Science. (n.d). Retrieved from https://en.wikipedia.org/wiki/Science

Witt, M., Carlson, J., Brandt, D. S., & Cragin, M. H. (2009). Constructing data curation profiles. *International Journal of Digital Curation, 4*(3), 93-103.

[About the authors]

Suntae Kim is a principal research engineer in the division of advanced information at the Korea Institute of Science and Technology Information. He works for the University of Science & Technology as an associate professor. He received his Ph.D. degree in library and information science from Chonbuk National University. Before his current appointment, he worked as a computer program developer at Linksoft which developed the NDSL and NOS. His research interests include research data management, research data platform, research data sharing, semantic web, metadata.