Challenges in Creating a Sustainable Generic Research Data Infrastructure

Richard Grunzke*, Ralph Müller-Pfefferkorn, Wolfgang E. Nagel
Technische Universität Dresden, Germany

Tobias Adolph, Christoph Biardzki, Anton Frank, Arndt Bode
Leibniz Supercomputing Centre of the Bavarian Academy of Sciences and Humanities, Germany

Anastasia Kazakova, Fidan Limani, Atif Latif, Anja Busch, Timo Borst, Klaus Tochtermann
ZBW - Leibniz Information Centre for Economics, Germany

Mathis Neumann, Nelson Tavares de Sousa, Ingo Thomsen, Wilhelm Hasselbring
Christian-Albrechts-Universität zu Kiel, Germany

Jakob Tendel, Hans-Joachim Bungartz, Christian Grimm
Verein zur Förderung eines Deutschen Forschungsnetzes e.V., Germany

Abstract

Research data management is of the utmost importance in a world where research data is created with an ever increasing amount and rate and with a high variety across all scientific disciplines. The challenges in creating a Generic Research Data Infrastructure (GeRDI) are discussed. GeRDI aims at providing a reference implementation for a federated research data infrastructure including interconnected individual repositories for communities and an overarching search based on metadata. The challenges involve a high variety of requirements, the management and development of the distributed and federated infrastructure based on existing components, the piloting within the use cases, the efficient training of users, and how to enable the future sustainable operation.

1 Introduction

This manuscript presents the challenges within the development of a Generic Research Data Infrastructure (GeRDI) [12]. Research Data Management (RDM) is defined as both the IT- and community-driven management of data that are input for and output of other scientific activities, such as publications, visualizations, surveys, experiments, measurements or simulations. The overall importance of research data for science, economy, and society and its appropriate management has recently been stressed by the *Commission High Level Expert Group on the European Open Science Cloud* [10] and the *Rat für Informationsinfrastrukturen* [7]. The following four domains are helpful to explicate the RDM concept and its tasks further: 1) Data can be generated or collected by a multitude of different IT systems. 2) Data have to be stored on appropriate storage management devices taking into account aspects such as size, location, device, speed, duration, and backup. 3) Data have to be curated, meaning both data and metadata have to be maintained. CRUD-workflows (creating, reading, updating, and deleting of data) have to be defined, including access rights and data versioning. 4) Data have to be made accessible for easy further use: common interfaces have to be provided and maintained to share, transfer, document, visualize, discover, and publish the data.

On top of these aspects, GeRDI will develop a reference implementation for a distributed and federated research data infrastructure, based on existing systems, that is both generic pertaining to the local repository software and specific with respect to the use cases of the individual scientific communities. Besides the repository software, a GeRDI node consists of hardware, additional software packages, and network infrastructure which are necessary to interconnect with other nodes. The entirety of such independent nodes will form the overall infrastructure, a virtual and distributed RDM system. The targeted users of this infrastructure are for example scientists that do not have a ready-made RDM solution at their disposal or users that wish to easily discover and access data on a Germany wide scale.

2 Complex Requirements and Software Management

It is essential to continuously be in contact with the prospective pilot users of the GeRDI infrastructure. This fundamentally enables to develop an understanding of their requirements and get continuous feedback.

*corresponding author, richard.grunzke@tu-dresden.de

To productively facilitate this, a substantial amount of effort has to be spent especially during the beginning but also during the whole project runtime. It was decided to work with use cases [1] and personas [3] as requirement analysis artefacts to formalize case studies. A major challenge in GeRDI is the quantity and variety of requirements. This involves a variety of scientific communities with heterogeneous research data and requirements and means to process and manage them. Two kinds of requirements are central here. First, generic requirements such as usability and performance are common across all communities. Second, domain requirements such as metadata handling and user interaction are highly specific to individual communities. Another major challenge is the highly differing communication basis and vocabulary between individual communities.

Another aspect of the project's pilot phase is the collaboration with existing research data repositories: They are heterogeneous concerning data structures, hardware setups, tools, and workflows used for data processing and regulated data availability for further usage. This requires a focus during the requirement development on the potential users. At the same time this adds complexity to the development process. Assessing the current situations of these diverse community members (workflows, data creation and processing, limitations) using interviews to identify personas and usage scenarios is challenging, especially because the development is distributed among several project partners, which requires appropriate coordination. The assessment must be iterative (while touching others project aspects like requirement specification and architecture), to move from user descriptions to a more formal, scenario-based requirement specification. This requires "a common language" (glossary, terms, definitions) for the user stories that abstract from community specific vocabulary. These stakeholder perspectives are the basis for the requirement analysis and specification phase to obtain a reference for defining the overall architecture (see Section 3), continuous testing and evaluation (see Section 4). An important goal is to define the scope of the software with respect to 1) stakeholder perspectives, 2) identification of discipline-specific and generic functionalities, 3) domain-specific data analytics, 4) access to research data repositories, and 5) non-functional requirements such as scalability, security, efficiency, and monitoring. This leads to additional challenges such as the assessment of existing solutions, prioritizing features (for the pilot phase), and a risk analysis (see Sections 3 and 4).

Due to the diverse requirements, a continuous software engineering process will be applied: a cycle of development, test, build, deployment, monitoring (and back to development) [8]. This allows for early acceptance testing (automated unit & integration tests) based upon the identified user behaviour, continuous user feedback, and quality improvement. It is vital to agree early on a complete development tool chain for development, testing, continuous software integration, and deployment. This also includes tools for issue and task management, user feedback, and documentation.

3 Implementation of the Federated Infrastructure

The practical realisation of the federated research data infrastructure includes the overall architecture, the structure and content of metadata, and the management of both data and corresponding metadata. The following fundamental assumptions are made. 1) Different communities with complex and varied individual requirements (cp. Section 2) are planned to be served. 2) GeRDI will be based on existing and quality assured software as a backbone for customizations, interoperability, maintenance and deployment. 3) Automation, such as for extracting and validating metadata and update deployment, will be incorporated wherever possible in order to keep the hurdles of usage and operation as low as possible.

The overall aims of creating the GeRDI research data infrastructure influencing the architecture are 1) interconnecting individual data repositories based on open standards by means of registries, protocols, metadata schemes and vocabularies, 2) consultation for and/or providing of repository software for creating individual community-specific data repositories, and 3) providing a data portal with semantic search capabilities operating with all connected data repositories. The GeRDI architecture will be defined based on these and the complex requirements (see Section 2), the policy to re-use existing and proven software components [2], and the available development resources. Various fundamentally important design decisions have to be made. For example: What should be generic and what domain-specific? What should be central (if anything) and what local? How should the infrastructure be made both evolvable and adaptable? How to ensure a convenient, secure, efficient, and continuous deployment of (parts of) the infrastructure? How to achieve interoperability with existing non-GeRDI data repositories? How to support data analytics and HPC capabilities? How to manage access, security, and privacy?

Based on the architecture, a major challenge is the identification and subsequent in-depth evaluation of relevant existing software systems for the possible reuse in GeRDI. Here, an example is the planned evaluation of the RDM repository framework KIT Data Manager [6, 9], based on the experiences in the MASi research data management project [11], as a candidate for the basis of the repository software within the GeRDI reference implementation. Based on the evaluation results, it will be estimated what effort is required to adapt these components for use in GeRDI. Finally, a careful decision has to be made what com-

ponents shall be re-used and adapted. This significantly influences the following work in many parts of the project.

An overarching and fundamental challenge is to create an infrastructure that fulfils the requirements of community-oriented research data management (see Section 2) to the highest degree possible under the conditions and restrictions of a both specific and generic approach. This aims at ensuring a low hurdle of entry and a high overall usability for users as non-functional requirements. Functional user-centric requirements that might need to be fulfilled include the following:

1. Handling of a high variety of both specific and generic metadata standards.
2. Semantic integration of distributed research data repositories towards an interoperable and interdisciplinary research data infrastructure.
3. Providing a trans-disciplinary semantic search.
4. Supporting automatic metadata validation.
5. Providing means to facilitate metadata curation.
6. Supporting the management of provenance information to enable reproducibility.

4 Deployment, Integration, and Evaluation across Distributed Centers

Two of the major security threats of web-based IT systems are security misconfiguration and the usage of known vulnerable components (cp. [4] A5 and A9). The available personnel to administrate each local node will vary and might also fail to update the GeRDI-software (A9) or mis-configure it (A5). Both might be threats to both local nodes and the overall security of the infrastructure. Therefore the deployment of updates has to be made as convenient as possible and mechanisms to put single nodes under quarantine have to be implemented by design.

The challenges concerning the operation of a distributed infrastructure heavily depend on the chosen infrastructure architecture paradigm: A more central approach might be easier in terms of coherence and diversity of the participating data centers, but also needs central organization, efficient communication channels, and funds to provide the maintenance of the central components. If this approach is chosen, further challenges consist in the management of releases and changes considering an arbitrary number of nodes and users. A distributed approach would in contrast make these challenges easier to meet, but has to take care of the bigger impact of lower homogeneity of both the local software and resources available. In this case several mechanisms have to be put in place, e.g. a technically enforced policy which makes sure that nodes do provide the necessary resources for their share of the jointly provided services (e.g. the addition of a petabyte of index data should be backed by according computing power and storage space).

As already stated in Section 2 the characteristics and therefore the IT know-how of the users will range over a broad spectrum. Whereas some communities will early adopt new features and make their workflows depend upon these, others will need assistance and training (cp. Section 5). This challenge has to be met in the early stage by a face-to-face approach, e.g. such as the PiCS workshops (cp. [5]). In a later phase the main challenge concerning community management will evolve into the refinement of users experiences. As stated in Section 2, these time-consuming activities are necessary, not only during the requirement analysis but during the whole project.

The technical part of the infrastructure evaluation partly depends on the architecture paradigm. After defining the appropriate key performance indicators, their distributed measurement and collection has to be managed efficiently. This is one aspect that has to be taken into account already during the design phase of both software and hardware. For punctual tests, such as load and performance tests, coordination mechanisms to get reliable and significant results need to take both technical (i.e. software components, dedicated infrastructure) and organizational (i.e. personal resources, well-documented procedures) considerations into account.

5 Training and Sustainable Operation

Apart from challenges with respect to requirements analysis, software engineering, and piloting, the GeRDI project faces the issue of sustainability in terms of a training framework, operational models, and funding.

The user base and administrators require appropriate training, so the developed infrastructure effectively used and deployed/maintained. One of the major challenges for the training team in a complex and long-lived software project such as GeRDI is to provide training and reference material for a wide spectrum of user expertise from novice to expert users. At the same time, this repository of material must be kept up to date with the evolution of the software over time to prevent the training from becoming stale or outright incorrect. In keeping with this project's open-source approach, the training material will also be made publicly available. Beyond the challenges of creating and maintaining a set of training materials, the training function also plays an essential role in collecting user feedback, both on the training material and on the GeRDI infrastructure itself. This feedback must be effectively communicated to the product development team to inform their work going forward. In that context, the training function will also be communicating new information into the project.

For a project-funded software development effort

aimed at long-term operations, the inevitable question is how to organise self-supporting operations when the project funding ends. This involves exploring aspects of hosting the significant amount of IT resources required, as well as models for financing this hosting and the ongoing development efforts to the mutual satisfaction of the participants. Our objective is to identify likely operational models and to rank their potential to ensure optimal sustainability for project operation into the future. The challenges here are to identify the needs of current and future stakeholders and to account for them while assessing the ability of the different operational models. This will require support for the correct and successful implementation of the selected operational model(s) to ensure the desired outcome.

At the end of the first three years of the GeRDI project, the wider roll-out of the infrastructure in terms of finance and funding will be prepared. One task will be to highlight the project's results and to propagate them in different communities as an infrastructure solution for interoperable research data management. For this purpose, workshops are planned to introduce GeRDI and to present the advantages of running a GeRDI repository node to fulfil the requirements of potential community partners.

6 Conclusion and Outlook

The challenges we presented can be classified as either technical or organizational:

Examples for technical aspects are architectural decisions, domain-specific requirements (i.e. RDM-related questions), and operational needs from the IT-specific part of the project. We see a lot of existing approaches, techniques, and tools we can use, learn from and develop further, and hope to contribute solutions to hitherto unresolved problems.

Organizational challenges such as community management and sustainability will necessarily open social and political dimensions. Whereas science is built upon critical review and a rational standard aiming at knowledge augmentation, social interactions sometimes follow a different logic. To quote the first point stressed by the Commission High Level Expert Group on the European Open Science Cloud (EOSC): "The majority of the challenges to reach a functional EOSC are social rather than technical" (cp. [10]). The critical resources necessary to meet this type of challenges consist of time and careful communication. One interesting aspect lies in the interconnectedness of the two domains: Technical solutions might ease some issues (cp. the often cited "Science 2.0") but can also raise social difficulties (such as technical interfaces for non-technical users).

Looking forward, we aim at the optimal balance between the social and technological dimensions of the challenges presented.

7 Acknowledgements

This work was supported by the DFG (German Research Foundation) with the GeRDI project (Grant No. BO818/16-1, GR4908/1-1, HA2038/6-1, NA711/16-1, TO199/15-1).

References

[1] A. Cockburn. *Writing Effective Use Cases*. Addison-Wesley, 2000.

[2] W. Hasselbring. "Component-Based Software Engineering". In: *Handbook of Software Engineering and Knowledge Engineering*. World Scientific Publishing, 2002, pp. 289–305.

[3] K. Baxter and C. Courage. *Understanding Your Users: A Practical Guide to User Requirements Methods, Tools, and Techniques*. Interactive Technologies. Elsevier Science, 2005.

[4] *OWASP Top 10 - 2013 - The Ten Most Critical Web Application Security Risks*. Tech. rep. The Open Web Application Security Project (OWASP), 2013.

[5] A. Frank et al. "In Need of Partnerships – An Essay about the Collaboration between Computational Sciences and IT Services". In: *Procedia Computer Science* 29 (2014). 2014 International Conference on Computational Science, pp. 1816–1824.

[6] T. Jejkal et al. "KIT Data Manager: The Repository Architecture Enabling Cross-Disciplinary Research". In: *Large-Scale Data Management and Analysis (LSDMA) - Big Data in Science*. 2014, pp. 9–11.

[7] *Enhancing Research Data Management: Performance Through Diversity*. Tech. rep. Rat für Informationsinfrastrukturen, 2016.

[8] W. Hasselbring. "Keynote: Continuous Software Engineering". In: *Software Engineering 2016*. Ed. by J. Knoop and U. Zdun. Vol. P-252. Lecture Notes in Informatics (LNI). Köllen Druck+Verlag GmbH, 2016, pp. 113–114.

[9] *KIT Data Manager*. `http://datamanager.kit.edu/`. Dec. 2016.

[10] *Realising the European Open Science Cloud*. Tech. rep. Commission High Level Expert Group on the European Open Science Cloud, 2016.

[11] R. Grunzke et al. "Towards a Metadata-driven Multi-community Research Data Management Service". In: *2016 8th International Workshop on Science Gateways (IWSG)*. 2016, accepted.

[12] *GeRDI Project*. `http://www.gerdi-project.de/`. Jan. 2017.