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. This paper especially discusses software engineering challenges stemming from creating a long-living software system. It 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 evolving requirements, the management and development of the distributed and federated infrastructure that are 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) [9] that aim at being run in a broad and long-term sense. 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* [7].

On top of these aspects, GeRDI will develop a reference implementation for a distributed and federated research data infrastructure, based on existing systems to form a virtual and distributed RDM system. The resulting service will be both generic to be widely applicable and of specific merit for the community partners who are involved. 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.

In Section 2 the GeRDI project with its aim to build a long-living software system for research data management is introduced. Section 3 presents related software engineering challenges with various aspects. The complex and evolving requirement analysis challenges are described in Section 4 while implementation aspects are detailed in Section 5. Section 6 describes challenges regarding the distributed deployment, integration, and evaluation. The challenges regarding training and the long-term sustainable operation are described in Section 7.

2 GeRDI as a long-living Software System

GeRDI is a 3 million Euro DFG-funded project in the program for Scientific Library Services and Information Systems (LIS) with a time frame of 3 years that started in November 2016. The project partners include the *Leibniz Information Centre for Economics*, the *Christian-Albrechts-Universität zu Kiel*, the *Leibniz Supercomputing Centre of the Bavarian Academy of Sciences and Humanities*, *Verein zur Förderung eines Deutschen Forschungsnetzes e.V.*, and the *Technische Universität Dresden*. The focus is on building an advanced software system that enables a long-term

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

infrastructure. GeRDI combines a broad level of expertises in advanced metadata and search technologies, software engineering, high-level operation, training concepts, sustainability, and research data management.

The overall aims of GeRDI are to 1) interconnect individual data repositories based on open standards by means of registries, protocols, metadata schemes and vocabularies, 2) consult for and/or supply 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. Research data management is a long-term mission considered on a par with the mission of classical libraries in safeguarding the scientific and cultural heritage, be it in digital of analogue form. To ensure the long-term availability and find-ability of research data, enabling software systems have to be designed, implemented, deployed, and maintained with longevity, maintainability, and sustainability as essential goals.

The involved applied research communities are highly diverse with life sciences, environmental sciences, digital humanities, and economics, beside others. An example use case consists of microscopes that create images in the range of GB/s. The resulting data needs to be annotated with metadata (data about data). This enables scientists to find specific sub sets of the vast amount of stored data at any point in time. Others include fishery data, economy publications, risk reduction data, and more. GeRDI facilitates research data management for such arbitrary communities while going a step further in enabling an overarching semantic search over any such domain specific data.

3 Software Engineering Challenges

A major challenge is to define the GeRDI architecture based on the complex requirements (see Section 4), the policy to re-use existing and proven software components [2], the available development resources, and the aimed-for longevity. Various fundamentally important design decisions have to be made while keeping in mind that these will only get more challenging over time with an increasing number of communities and the resulting more complex requirements. Examples are:

Which components should be central (if any) and which 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 best achieve interoperability with existing non-GeRDI data repositories? How to interface with data analytics and HPC capabilities? How to manage access, security, and privacy? How can this software system be run for 20 or 30 years? To what extend can it be adapted according to new requirements? In what way can maintainability be ensured? How can it deal with constantly changing short-living hardware systems? How should software migration requirements be defined in this regard? Can the microservices concept be utilized to create a modular structure with easier to exchange individual parts? How should new research data repositories be integrated on-the-fly? In what way can extremely heterogeneous metadata be integrated and utilized to facilitate find-ability? How can the development and operation be tightly integrated with each other to more quickly enable a more advanced system? Which delivery and/or deployment strategies are best employed to better reach new researchers? How to integrate new automated quality management technologies to increase the overall software quality? How can interfaces offer stable functionality while at the same time being able to evolve? While the main focus of GeRDI is to facilitate research data management especially in regard to searching, to what extent is the integration of added value functionality such as HPC/analytics functionality advisable? How to deal with challenges that are especially present in distributed environments such as latency, synchronisation, and bandwidth? Although GeRDI in principle aims at being a generic service, to what extent could/should/must domain specific functionality be included? How to deal with dynamic data sources, such as databases, in a reproducible way? To what extent can automatically deployable repositories pre-filled with clearly defined content be used for testing in a distributed environment such as GeRDI? How can a project best deal with the situation that an utilized underlying complex library or technology is not developed any more? How can the developed code be managed, maintained, and extended in a long-term sustainable way?

4 Complex Requirements and Software Management

GeRDI aims at providing a generic infrastructure for with various communities (see Section 2). This results in highly complex, heterogeneous and evolving requirements with more communities becoming part of GeRDI. Due to these points, it is a necessity to continuously stay in close contact with the communities during the whole project lifetime. This fundamentally enables to develop and keep an understanding of their evolving requirements and get continuous feedback. To productively facilitate this, a substantial amount of effort must be spent especially during the beginning but also during the whole project runtime. We need to keep our requirement analysis setup lean and extendible to deal with the complex and evolving requirements. To handle these challenges, we separate our questionnaire in a generic and a domain specific part and work with use cases [1] and personas [3] as requirement analysis artifacts. Since our scientific communities have heterogeneous research data, require-

ments and means to process and manage them, we see two main kinds of requirements. 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. GeRDI aims at handling both kinds of requirements. Thus, we must gather generic requirements in respect to usability and extend our methods in respect to the gathering of metadata requirements, e.g. with the means of technical analysis methods. This is needed as not every community is deeply or at all aware of possible metadata and can give us appropriate information about it. Another major challenge is the highly differing communication basis and vocabulary between individual communities. This challenge is met by creating interview guidelines, the careful analysis of their results, and the increasing experience of community managers to efficiently interact with their communities.

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) [5]. This allows for early acceptance testing (automated unit and 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.

5 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 4) 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.

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 [4, 6], based on the experiences in the MASi research data management project [8], 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 components shall be re-used and adapted. This significantly influences the following work in many parts of the project and the long-term behaviour and characteristics of the system.

6 Deployment, Integration, and Evaluation across Distributed Centers

The challenges concerning the long-term operation of a distributed infrastructure heavily depend on the chosen infrastructure architecture paradigm: A more central approach might be easier in terms of maintainability and coherence between the participating data centers. With a more central organization the responsibility is concentrated in fewer organisations and with sufficient funding the longevity of the system is facilitated. If this approach is chosen, further challenges consist in the management of releases and changes considering an arbitrary number of connected nodes and users. A more distributed approach would in contrast make the system more modular and, thus, potentially easier to maintain the individual components, but it would also result in a bigger impact of lower homogeneity of both the local software and resources available. This is especially challenging in the long-term with independent organisations in various federal states with differing funding sources, where specific parameters might change over time.

The technical part of the infrastructure evaluation partly depends on the architecture paradigm. While defining the appropriate key performance indicators, it is especially important to keep in mind that the number of users and scalability requirements will likely increase over time. 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.

7 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.

8 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. [7]). 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.

9 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] 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.

[5] 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.

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

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

[8] 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.

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