# How to Choose a Research Data Repository Software? Experience Report

Nina Buck[1],Volodymyr Kushnarenko[1], Björn Schembera[1], Mona Ulrich[2], Heinz Werner Kramski[2], Andreas Ganzenmüller[1], Jan Hess[2], Alexander Holz[2], André Blessing[4], Pascal Hein[3], Kerstin Jung[4], Nicolas Schenk[2], Claus-Michael Schlesinger[3], Thomas Bönisch[1], Roland S. Kamzelak[2], Jonas Kuhn[4], Gabriel Viehhauser[3]

[1]High-Performance Computing Center Stuttgart (HLRS), University of Stuttgart;
[21]German Literature Archive Marbach (DLA);
[3]Institute for Literary Studies / Department of Digital Humanities (ILW), University of Stuttgart;
[4]Institute for Natural Language Processing (IMS), University of Stuttgart

In the age of digital transformation, scientific and social interest for data and data products is constantly on the rise. The volume as well as the variety of digital research data is increasing significantly. This raises the question about the management of this data. For example, storing data so that it is presented transparently, freely accessible and subsequently available for re-use to comply with good scientific practice. Research data repositories provide solutions to these issues and foster compliance with the FAIR principles.

Considering the variety of available software products, it is sometimes difficult to identify a fitting solution for a specific use case. This paper shares our experiences during the process of assessing, choosing and implementing a research data management repository. We provide a brief reflection about standard repository software in contrast to in-house development, describe several software solutions and their features, show software testing results and provide recommendations for assessing and choosing a suitable solution. This paper is aimed in particular for researchers, projects and institutions searching for a suitable software solution to set up and run a repository.

## 1 Introduction

Within the SDC4Lit project (Science Data Center for Literature)[1] a sustainable repository for net literature and born digitals aims to be built. For this purpose, a suitable open-source research data repository software is needed. Considering the variety of repository

Publiziert in: Vincent Heuveline, Nina Bisheh und Philipp Kling (Hg.): E-Science-Tage 2023. Empower Your Research – Preserve Your Data. Heidelberg: heiBOOKS, 2023. DOI: https://doi.org/10.11588/heibooks.1288.c18075 

1 https://www.sdc4lit.de

software nowadays, it is sometimes difficult to choose the right one. In this paper we want to share our experience, how to assess a repository to your needs. The paper starts with a brief reflection about a standard research data repository software in contrast to in-house development (section 2), explains the role of requirements for choosing or developing repositories (section 3), introduces several software solutions and their features (section 4), shows some of our software testing results (section 4) and provides recommendations on how to choose a suitable repository software pertaining to a use case based on our experience (section 5). In addition to this paper, we provide a catalogue with general requirements regarding repository software and specific requirements pertaining to the use case in SDC4Lit. This paper and the catalogue can be used as a starting point for other groups and projects who plan to set up and run a research data repository.

## 2 Standard software or individual solution

Setting up a repository raises the question whether already existing software solutions (standard software) should be used or if it is necessary to develop a new and individual solution from scratch.

The use of a standard software has the advantage that further development and support are more extensively ensured than in the case of individual development, especially through the larger user community of the standard software (Winkler 2008). A ready-to-use solution is usually offered as open source and free of charge and has many already integrated standard functions. A usage of such a software is usually not complicated and quick start is also possible, even if some local adjustments might be necessary. To have it even easier, some proprietary, tailor-made solutions available on the market can be considered as well. They are very convenient, flexible and easy to use, but are usually tied to a pricing model, which can sometimes be unacceptable.

Development of own repository applications is recommended only in cases when none of the offered software solutions can meet individual requirements (ibid.). It is a customised software that is ideally adapted to the local requirements. However, the development of such software is very time-consuming and cost-inefficient, so using and customizing standard software whenever possible is a clear advantage.

## 3 Requirements

The technical implementation and operation of a repository necessitates selecting suitable software packages according to a number of requirements. These requirements can be defined based on the use cases and workflows of individual projects or institutions. Besides common requirements like the assignment of persistent identifiers (PID), the implementation of a specific metadata model or the availability of application programming interfaces (API), which are available in almost all repository software, sometimes there are also very special requirements such as rendering of directory structures or the capability to operate extremely large files, which not every software is able to fulfill.

There is no common checklist for the selection of a software solution. It always depends on the individual requirements of the institution running the repository. However, there is already a published article that describes possible criteria and their relevance for the selection of repository software (Axtmann et al. 2021a) as well as a catalogue of general requirements for the repository software (Axtmann et al. 2021b) which both can be used as a guidance. In addition to this we provide a catalogue of special requirements for the repository for net literature and born digitals as a supplement to this paper (Buck et al. 2023), which also includes a mechanism to create a ranking of repository software based on fulfillment of the requirements.

# 4 Standard Software overview

There are a number of standard repository software. Some are widely spread, making their names already known, others are less common or used in only one or several countries. An overview of standard currently used repository software can be found here:

- Registry of Research Data Repository[2]
- Directory of Open Access Repositories (Open DOAR)[3]

Some popular free of charge repository solutions are DSpace[4], Dataverse[5], Fedora[6] and its Framework Islandora[7]. Two others – MyCoRe[8] and Invenio[9] – have been established in German-speaking countries. The list of available and also free of charge repository software can be also extended with Samvera[10], OPUS[11], EPrints[12], Software Heritage[13], CKAN[14], AtoM[15], LibreCat[16], etc. Below follows a brief overview of the software and their features, which were tested as part of the project SDC4Lit.

The overview and results described in this paper are based on the research about repository software made in 2020. Therefore the current version of the software may offer a different feature set as described below. However, the described procedure for selecting a repository software based on individual requirements is valid on general principle.

2 `https://www.re3data.org/metrics/software`; *Last accessed on May 23rd, 2022.*
3 `http://v2.sherpa.ac.uk/view/repository_visualisations/1.html`; *Last accessed on April 25th, 2022.*
4 `https://duraspace.org/dspace`; *Last accessed on April 25th, 2022.*
5 `https://dataverse.org`; *Last accessed on April 25th, 2022.*
6 `https://duraspace.org/fedora`; *Last accessed on April 25th, 2022.*
7 `https://www.islandora.ca`; *Last accessed on April 25th, 2022.*
8 `https://www.mycore.de`; *Last accessed on April 25th, 2022.*
9 `https://invenio-software.org`; *Last accessed on April 25th, 2022.*
10 `https://samvera.github.io/introduction.html`; *Last accessed on May 27th, 2022.*
11 `http://www.opus-repository.org`; *Last accessed on May 27th, 2022.*
12 `https://wiki.eprints.org/w/Introduction`; *Last accessed on May 27th, 2022.*
13 `https://archive.softwareheritage.org`; *Last accessed on May 27th, 2022.*
14 `https://ckan.org`; *Last accessed on May 27th, 2022.*
15 `https://www.accesstomemory.org/de`; *Last accessed on May 27th, 2022.*
16 `https://github.com/LibreCat/LibreCat`; *Last accessed on May 27th, 2022.*

## 4.1 DSpace

DSpace was originally developed for document management in 2002 at the Massachusetts Institute of Technology (MIT) and the research department of Hewlett-Packard (HP) and is currently managed by the non-profit organisation DuraSpace. Various service providers[17] offer commercial development of the code. DSpace is used by more than 1000 organisations, making it the most widely used standard software for repositories, which speaks for a large and active community.

DSpace is free, easy to install and fully customisable. DSpace supports the OAI Protocol for Metadata Harvesting OAI-PMH, comes with a Solr-based search mechanism and internal checksum verification. Also Shibboleth integration is possible via a plug-in.

DSpace 6.0 was investigated. Late summer 2021 a new version DSpace 7.0 was released. It has a completely new interface and some new features.

How DSpace works can be tried at a demo version[18]. Several login credentials are provided, so it is not necessary to create own account.

## 4.2 Fedora

Fedora was developed at the University of Virginia and Cornell University. The project is led by the Fedora Leadership Group and overseen by the non-profit organisation DuraSpace.

Fedora is a robust, modular and freely available repository software for managing digital content. It is primarily used in libraries, universities and other research and academic institutions as a data repository for document servers. Fedora enables access to very large and complex digital collections. It features very high flexibility and supports all metadata schemas. It has a RESTful API, checksum verification and enables Shibboleth integration. However, Fedora is only a minimal environment that needs to be significantly extended for productive use (see section on Islandora).

Fedora 5.1.0 was investigated. Meanwhile Fedora 6 was released. Unfortunately, there was no demo instance offered.

## 4.3 Islandora

Islandora was originally developed at the University of Prince Edward Island (UPEI) in 2009 and is now used by over 300 institutions.

Islandora is an open source software framework based on Fedora and content management system Drupal. It can be extended with several modules developed by the Drupal community. The software comes with a default configuration, which provides basic functionality.

17 `https://duraspace.org/dspace/resources/service-providers`; *Last accessed on April 25th, 2022.*

18 `https://demo7.dspace.org`; *Last accessed on May 23rd, 2022.*

Islandora is widely used and has a strong community. However, the software product is very complex and consists of many microservices. The installation went not smoothly and a high maintenance and operation effort was estimated.

There is no native possibility to transfer a directory tree on a hard disk into a repository. The directory structures must be simulated via "member of" relationships.

Islandora 7 and 8 were tested. Nowadays version 9 is available. A demo version[19] with provided login credentials could also be tried out.

## 4.4 Invenio

Invenio was originally developed at CERN in 2002 as a document server. It has been further developed and nowadays consists of three products[20]:

- InvenioRDM – a repository/document management platform,
- InvenioILS – an integrated library system,
- Invenio Framework – a code library to build large-scale information systems such as InvenioRDM and InvenioILS.

Invenio is an open access software and is designed to work with huge amount of data as well as large datasets. Metadata from various sources can be integrated. Invenio provides a PID-store and resolver that allows you to use a preferred PID scheme to identify records. Invenio uses Elasticsearch as a search engine.

Two Invenio instances were installed and tested: Invenio Framework and InvenioRDM. Despite numerous tutorials, the installation went not without problems. Configuration and extension of the repository was time-consuming and required additional programming.

At the time of testing, InvenioRDM was not a finished product and had only a few functionalities. The operation of this software was done mostly via command line. A Web-Interface was not fully available during the testing.

Invenio Framework is very modular and allows a wide range of applications to be served. The operation of this software via web interface was very limited. File upload was done via the command line. Mapping of the data hierarchy was also possible, as paths, but there was no direct implementation on the web interface.

Only for InvenioRDM is a demo version[21] available.

19 `https://sandbox.islandora.ca`; *Last accessed on May 23rd, 2022.*
20 Invenio Homepage, `https://invenio-software.org`; *Last accessed on April 25th, 2022.*
21 `https://inveniordm.web.cern.ch`; *Last accessed on May 23rd, 2022.*

## 4.5 MyCoRe

MyCoRe was developed at the University of Duisburg-Essen. The office is located at the Regional Computer Centre of the University of Hamburg. MyCoRe is mainly distributed within Germany. The community is not very large, but is always ready to help.

MyCoRe framework provides all basic functions of document and publication servers. Own web applications can be developed through adaptations. An integrated image viewer is provided. Metadata models are customisable and extensible. Persistent identifiers ensure permanent access to the data.

MIR (MODS Industrial Repository) is an application that can be installed out-of-the-box. It provides all typical repository functions and can be used productively immediately. Adaptations are only foreseen for the layout and web content.

MyCoRe LTS 2019.06.04 was tested. The installation is slightly complicated as there are no step-by-step instructions and many components such as Solr, Apache Tomcat, etc. have to be configured by oneself. The web interface can be fully customised. During the upload of data it is possible to drag and drop entire directories and also lists of directories and files into the corresponding place and keep the original structure of the data.

Every year a new Long Term Support (LTS) version is released. A demo version[22] with several login credentials is also available.

## 4.6 Dataverse

Dataverse originates from the Institute for Quantitative Social Science (IQSS) in a collaboration with the Harvard University Library and Harvard University Information Technology. It is an open source software, has a supportive developer community and is distributed worldwide.

Dataverse repository software has two types of "data containers": dataverses and datasets. Dataverses (logical dataverses within a Dataverse installation) are the organisational structure of the repository. Datasets are the organisational structure below dataverses. They contain files and associated descriptive metadata. Datasets are nestable and can imitate directory structures. All files inside of datasets can be represented as a directory structure via the activated Tree-view.

Dataverse supports Search, Data Access, Metrics and Native APIs, as well as SWORD, OAI-PMH and Solr. It provides versioning and data citation. Authentication is possible via local accounts as well as via Shibboleth, ORCID, Google or GitHub. Roles and rights can be defined for each dataverse or dataset. The File Previewer and some other tools can be additionally integrated, which provides an opportunity to extend the functionality of the repository.

22 `https://www.mycore.de/mir/content/index.xml`; *Last accessed on May 23rd, 2022.*

Software versions since 5.3 were tested. Following the installation guide[23], it is quick and easy to install Dataverse. Upgrading to the next versions is not difficult either.

A demo instance of Dataverse[24] is also available.

## 4.7 Testing and Results

At first glance, there are no significant differences between repository software solutions. Therefore, the search for suitable software should be more precise and targeted explicitly at the area of application. But how should it be done exactly?

For a more detailed investigation, it is recommended to install an eligible software and evaluate it according to the collected requirements. On the one hand, you can see how the installation proceeds and how the further operation of the repository could look like. On the other hand, the specified requirements can be precisely tested to see whether they are fully, partially or not fulfilled.

# 5 Own experience

SDC4Lit was created with the aim to reflect on the requirements that digital literature places on its archiving, research, and mediation, and to implement appropriate solutions for a sustainable data life-cycle for literary research and mediation in the long term. In the course of this a long-term repository for digital literature is to be established. It should serve as a central repository for genuinely digital literary materials, literature on the net and parts of the author's inheritances, born-digitals. Literature on the net collection includes archived websites, literary blogs or online magazines related to the modern German literature. This collection consists of about 500 sources and results in a data volume of about 9TB with annual growth of 1TB. Born-digitals collection consists of about 75 inventors and represents a total data volume of about 2.8TB, stored in different formats and distributed over about 2000 data carriers.

In order to find the most suitable software solution for SDC4Lit, several potential use cases and resulting requirements were collected. These requirements (Buck et al. 2023) are the part of the catalogues of requirements (Axtmann et al. 2021b) mentioned in section 3. Some of the most essential requirements in SDC4Lit were the rendering of directory structures and possibility to allocate a persistent identifier for each file in a dataset, which are not common for research data repositories.

Based on requirements, it is now clear what should be included in a software package. At first, documentations of software products were examined to see if one or another requirement might be fulfilled. Since not all repositories were up to date, we limited our search to DSpace, Dataverse, Fedora, Islandora, MyCoRe and Invenio. All these

23 `https://guides.dataverse.org/en/latest/installation/index.html`; *Last accessed on June 1st, 2022.*

24 `https://demo.dataverse.org`; *Last accessed on May 23rd, 2022.*

repository solutions are widely spread and have a big community. They are continuously developed and improved. But not all of them could fulfill our requirements. Thus the choice had to be reduced again.

To evaluate the installation process and all requirements, Dataverse, MyCoRe, Islandora and Invenio were installed. Afterwards each software was weight up by assigning points for fulfillment of requirements. In the end, Dataverse met our essential requirements and suited slightly better than other candidates, therefore it was chosen as a repository software for the SDC4Lit data.

# 6 Summary

The variety of open source repository software is large. Each product has many features, most are the same or similar, but some are only available in one or another software. If you want to build your own repository, you need to consider many aspects, because developing a repository from scratch turned out to be difficult, and good repository products are already available. During our project, we tested several repository software and created a list of requirements that is aligned to our data, that helped us to select the most suitable software for our data. For this reason, collected references and catalogues of requirements presented in this paper can serve as a starting point and decision-making guide for choosing a proper repository software for project-specific data, regardless of the discipline from which the data originate.

# Acknowledgements

This research was done in scope of the SDC4Lit project, funded by the Ministry of Science, Research and the Arts Baden-Württemberg (MWK).

# References

Axtmann, Alexandra, Felix Bach, Jonathan Bauer, André Blessing, Thomas Bönisch, Nina Buck, Holger Gauza, et al. 2021a. "Kriterien für die Auswahl einer Softwarelösung für den Betrieb eines Repositoriums für Forschungsdaten". *Bausteine Forschungsdatenmanagement,* number 3: 14–26. DOI: https://doi.org/10.17192/bfdm.2021.3.8348. https://bausteine-fdm.de/article/view/8348.

———. 2021b. *Kriterien für die Auswahl einer Softwarelösung für den Betrieb eines Repositoriums für Forschungsdaten.* DOI: https://doi.org/10.5281/zenodo.5562885.

Buck, Nina, Volodymyr Kushnarenko, Björn Schembera, Mona Ulrich, Heinz Werner Kramski, Andreas Ganzenmüller, Jan Hess, et al. 2023. *How to choose a research data repository software? Experience report. Table of requirements.* DOI: https://doi.org/10.5281/zenodo.7656573.

Winkler, Marco. 2008. “Langzeitarchivierung von Online-Publikationen digitaler Repositorien”. Diplomarbeit, Fachhochschule Potsdam.