Efficient Data Curation by Automating Workflows

Florian Fritze [1,2], Dorothea Iglezakis [1,2,*], Sarbani Roy [1,2], Björn Selent [1,3,*], Karoline Weinspach [2]

[1] Research Data Competence Center, University of Stuttgart;
[2] University Library, University of Stuttgart;
[3] Technical Information and Communication Services, University of Stuttgart;
[*] Corresponding author: fokus@izus.uni-stuttgart.de

Data curation results in high-quality data description, but takes time and effort. To still enable an intensive data curation process at the institutional data repository *DaRUS*, different approaches support and automate the process: to support correct and standardized input as early as the metadata entry stage, the researchers are assisted by interfaces to registries like ORCID for authors, ROR for institutions, to terminology services for a topic classification and to the institutional research information system for research projects. Automatic checks for keywords, URLs and funding information are embedded in a REST API named pubWorkflow to help the curation team and manage the publication workflow. An integration with easyReview is planned to also support a scientific quality check.

All integrations and tools build on the workflow engine and the controlled vocabulary support of Dataverse, the underlying repository platform.

Keywords: Data Curation, Metadata, Data Repository, Automation

Published in: Vincent Heuveline, Philipp Kling, Florian Heuschkel, Sophie G. Habinger, and Cora F. Krömer (Hrsg.): E-Science-Tage 2025. Research Data Management: Challenges in a Changing World. Heidelberg: heiBOOKS, 2025. DOI: https://doi.org/10.11588/heibooks.1652.c23927 (CC BY-SA 4.0).

1 DaRUS – The Data Repository of the University of Stuttgart

The University of Stuttgart's research data repository *DaRUS* is based on the open source software Dataverse[1]. Dataverse originates from the Institute for Quantitative Social Science (IQSS) of Harvard University which actively maintains the project and leads its ongoing development. A strong international community comprising 125 worldwide institutional installations and the Global Dataverse Community Consortium (GDCC)[2] vouches for Dataverse being a lasting solution in terms of research data management. *DaRUS* allows researchers of the University of Stuttgart to share their data with both the scientific community and public while keeping full control of the data. Dataverse's features include the possibility to organize and structure data and access to it, to enrich the data with descriptive metadata and eventually to publish it to the world. From the onset the project aimed to provide a tool for research data management that strongly adheres to the *FAIR* principles, i.e. to make the data *F*indable, *A*ccessible, *I*nteroperable and *R*eusable (Wilkinson et al. 2016).

To better meet the needs of the University of Stuttgart, individual adaptations to the upstream project have been implemented for *DaRUS*. Besides minor changes to the look and feel of the software and the handling of custom mime types of uploaded files, the changes make extensive use of Dataverse's inbuilt extension methods in order to add supplementary functionality. These methods are in particular the option to alter and add the schemata defining the metadata fields which are used to enrich the uploaded data, the possibility to hook external tools to Dataverse, the linking of external vocabularies which can be searched for suitable entries to fill the metadata fields and a workflow mechanism which can be used to call external actions prior to publishing and after successful publishing res.

In *DaRUS* external tools are used to preview a wide range of mime types, such as images, videos, tabular data or text data. Even zip-archives can be inspected by an external tool previewer. Individual discipline specific metadata blocks have been developed in Stuttgart and added in order to enter pinpoint information for data stemming from engineering or chemistry research. These metadata blocks basing on schemata and standards like *EngMeta* (Schembera and Iglezakis 2020; Seeland 2020), *EnzymeML* (Lauterbach et al. 2023) and *CodeMeta* (Jones et al. 2017; Iglezakis 2023), proved to be useful and thus have been adopted by other Dataverse installations, too. Furthermore, a block has been added which helps to describe settings and requirements to run simulations and experiments. The block containing relevant information for citation has been extended by project and funding information. These fields are already linked to an external vocabulary in order to assist with filling the data. More detailed information about this feature will be given

1 https://www.dataverse.org; *Visited on March 28, 2025.*
2 https://www.gdcc.io; *Visited on March 28, 2025.*

in section 2.1. Finally, *DaRUS* also uses an external workflow prior to publishing which will be described in section 2.2.

The combined strength from a solid upstream code base and a vibrant community makes Dataverse a sound tool for scientific RDM in general and a valued assistant to the scientists of the University of Stuttgart in particular. Since its introduction in 2019 *DaRUS* gained significantly in popularity as can be seen in Figure 1(a). A growing number of institutes of the University of Stuttgart own their Dataverse collection in *DaRUS* and keep creating an increasing number of datasets. Alongside with the increasing number of created datasets the number of published datasets grew constantly over the last years as shown in Figure 1(b). Thus, the repository plays a key role in both enhancing open science and accrediting the work involved with generating the data.

The researchers also notably recognize the added value of data containing rich and correct metadata and are ready to provide all necessary information. Nevertheless, it still means an additional effort to fill datasets with this metadata and not always is it possible to include dataset creation in procedural workflows. Numerous metadata also leads to an additional workload on the data curation level. Thus, in order to overcome this side effect of the increased use of *DaRUS*, several means have been introduced into *DaRUS* to ease and automate the workflows involved with dataset management for both the researchers and the curators.

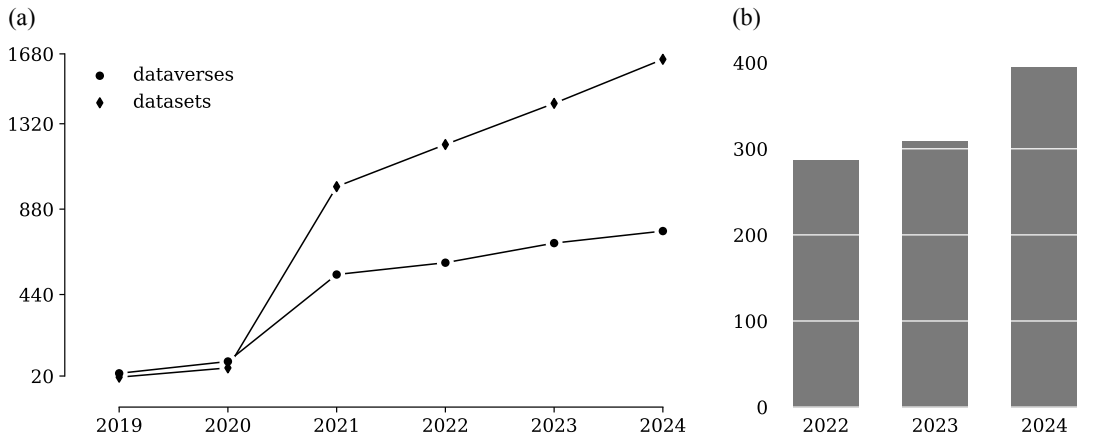


Figure 1: *DaRUS* usage (a) and number of published datasets (b).

1.1 The Publication Process

The publication process for datasets in *DaRUS* consists of several stages which involve different actors and actions. An overview is presented in Figure 2. The process starts with an author creating an initial dataset draft either by using the GUI or the API of *DaRUS*. At this stage only a limited number of metadata is required in order to do so successfully and the dataset is unconditionally editable. Once the author is inclined to publish the

dataset a Content Review process is triggered and a curator is notified. The curator is usually a peer of the author and the review at this stage is meant to contain a scientific discipline-specific survey of the data. The data is locked. If the data needs some form of alteration, the curator will unlock the dataset and return a review report with remarks to the author who will need to incorporate the changes. If the data passes the content review, the curator will start the publication process that involves the University's library staff for a formal and semantic review. The dataset is locked at this stage again until it either is returned to the author for major changes or unlocked for small changes by the curation team. If all parties involved are content with the review's outcome, the dataset is finally released and sent to the university bibliography, its PID is registered, and a fixed version of the dataset is published.

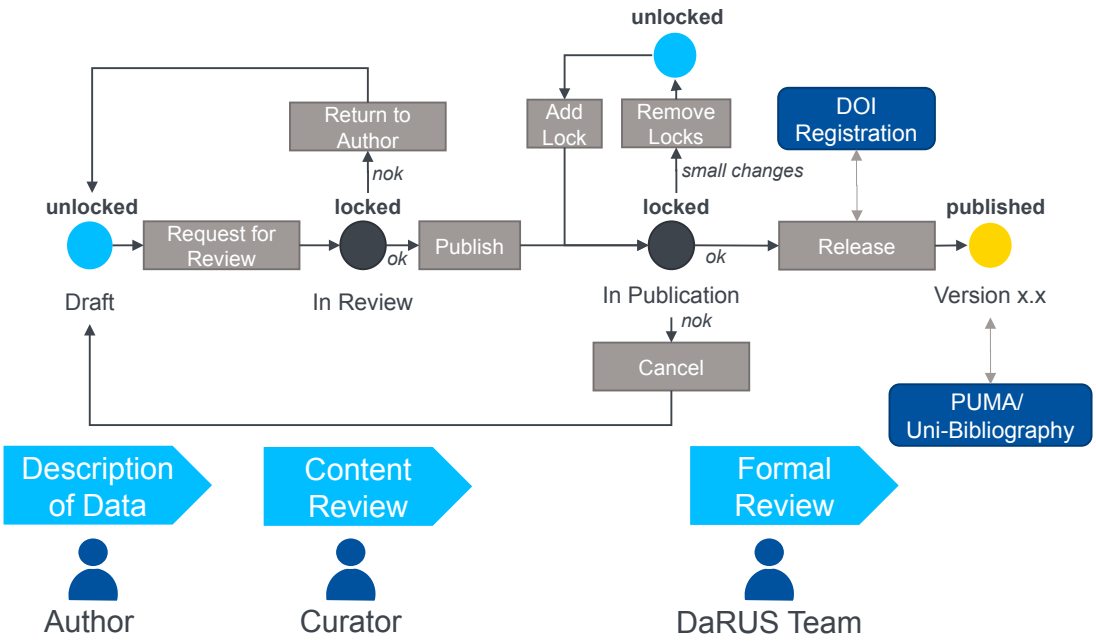


Figure 2: Publication workflow of DaRUS.

1.2 Data Curation at DaRUS

In this formal review, each dataset on *DaRUS* is checked for comprehensibility, reusability, interoperability and archivability[3]. To be *comprehensible*, the content and structure of the dataset should become clear without any further external information. The authors should describe the meaning of the files of the dataset, explain used abbreviations in file names, and make clear, how the data or code can be used. This information is expected in human-readable form in the description of the dataset and/or in a README file. The curators also look for and correct typos and enhance the readability. Figure 3 shows an example of a considered good dataset description.

3 https://www.izus.uni-stuttgart.de/en/fokus/darus/qualitycontrol/; *Visited on March 28, 2025.*

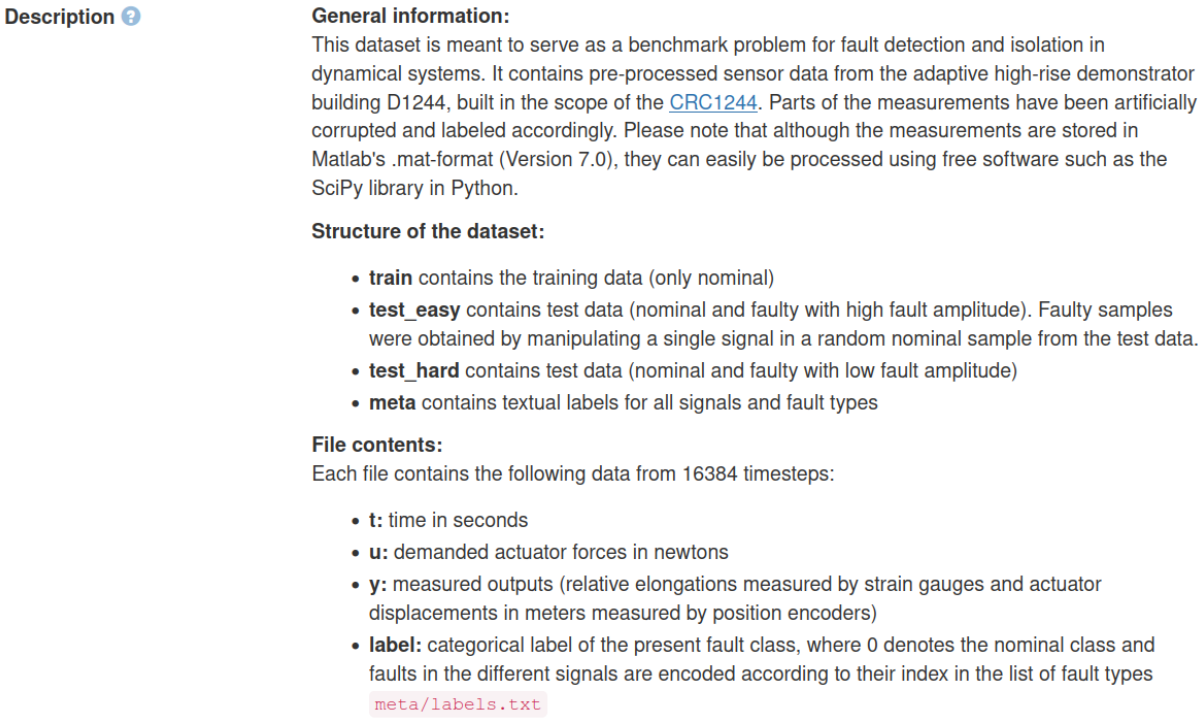


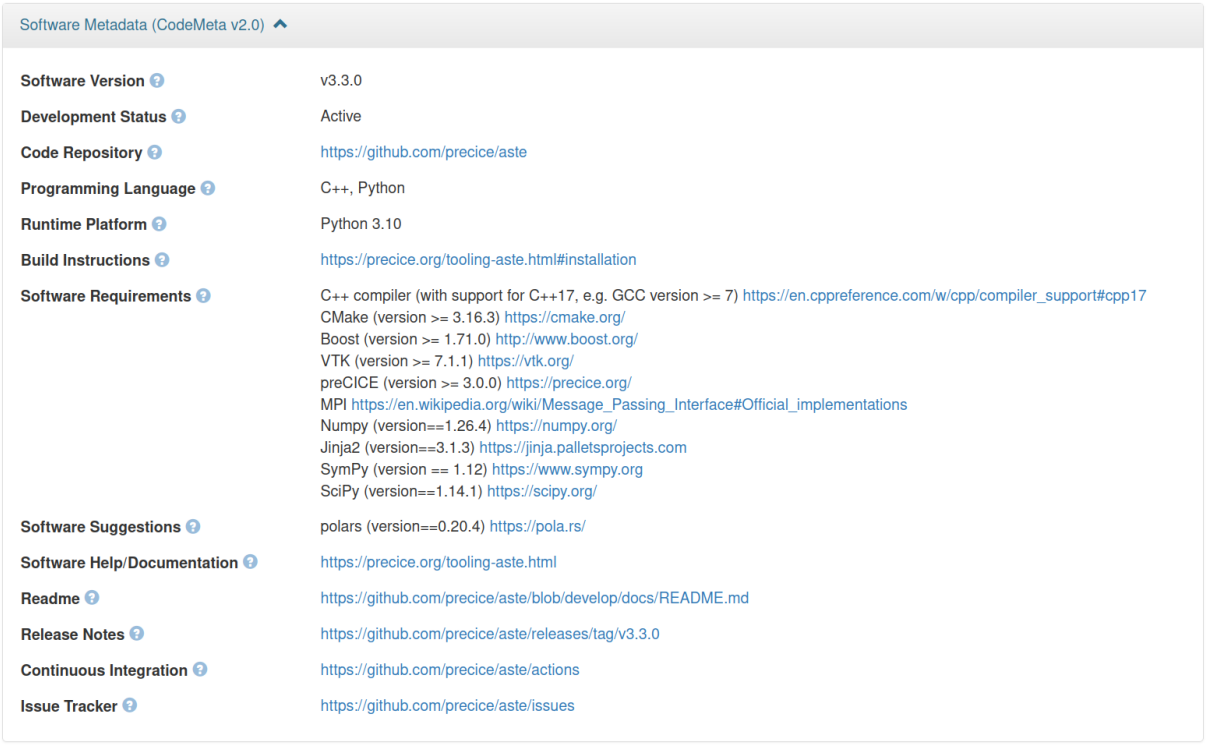
Figure 3: Exemplary dataset description.

To be *reusable*, a description of the tools and methods used to generate the data or necessary to open the data is essential. The latter is especially important for data in a proprietary data format that can only be read with specific software. The process metadata block basing on EngMeta provides the necessary metadata fields, for which completion is requested if applicable.

If the dataset contains primarily software or scripts rather than data, the authors are encouraged to provide software specific metadata via a CodeMeta[4] based metadata block. Metadata fields are the necessary dependencies with their version, CPU or memory requirements or links to documentation for example. This information is supposed to ease reuse of the software or to make the code run even after a few years have passed and the original users are unavailable (see Figure 4 for an example).

In addition, *DaRUS* users are assisted in converting their data in an open format, especially to transform tabular data in a specific csv format that enables Dataverse to transform the data in an archival format to make it available in different formats and offer the aforementioned preview functionalities for the data.

4 https://codemeta.github.io/index.html/; *Visited on March 30, 2025.*



Software Metadata (CodeMeta v2.0)

Software Version	v3.3.0
Development Status	Active
Code Repository	https://github.com/precice/aste
Programming Language	C++, Python
Runtime Platform	Python 3.10
Build Instructions	https://precice.org/tooling-aste.html#installation
Software Requirements	C++ compiler (with support for C++17, e.g. GCC version >= 7) https://en.cppreference.com/w/cpp/compiler_support#cpp17 CMake (version >= 3.16.3) https://cmake.org/ Boost (version >= 1.71.0) http://www.boost.org/ VTK (version >= 7.1.1) https://vtk.org/ preCICE (version >= 3.0.0) https://precice.org/ MPI https://en.wikipedia.org/wiki/Message_Passing_Interface#Official_implementations Numpy (version==1.26.4) https://numpy.org/ Jinja2 (version==3.1.3) https://jinja.palletsprojects.com SymPy (version == 1.12) https://www.sympy.org SciPy (version==1.14.1) https://scipy.org/
Software Suggestions	polars (version==0.20.4) https://pola.rs/
Software Help/Documentation	https://precice.org/tooling-aste.html
Readme	https://github.com/precice/aste/blob/develop/docs/README.md
Release Notes	https://github.com/precice/aste/releases/tag/v3.3.0
Continuous Integration	https://github.com/precice/aste/actions
Issue Tracker	https://github.com/precice/aste/issues

Figure 4: Exemplary metadata documentation of a research software set.

To be *interoperable*, the metadata should be linked to persistent identifiers (ORCID[5] for authors and ROR IDs[6] for organizations, DOIs for links to text publications and other datasets) and use controlled vocabularies for subject indexing like keywords and topic classification.

The curators also ensure that all used URLs resolve at least at the time of publication and that the contact information is long-term reachable.

Each dataset is curated by two different persons, one from the library publication team and one from the data management team. The curation effort for each dataset including all communication with the authors range between 10 minutes and several hours, depending on the state of completeness of the available metadata and the experience of the authors. The effort is rewarded by high-quality data sets and the authors' quick learning. Data curation serves as RDM consulting through the back door, so to speak. However, as the volume of publications increases, the workload on the curation team also rises steadily.

5 https://orcid.org; *Visited on March 30, 2025.*
6 https://ror.org; *Visited on March 30, 2025.*

2 Automating workflows

Two different approaches are used to keep the curation effort as low as possible and thus make the publication workflow scalable: On the one hand, support for authors through interfaces to external PID and vocabulary services, and on the other hand, automated checks to support curators and reviewers.

2.1 User Input Support by Interfaces

Efficient metadata management is crucial for ensuring the discoverability, interoperability, and consistency of research data. To achieve this, *DaRUS* integrates third-party vocabulary and persistent identifier (PID) services using Dataverse's flexible external vocabulary support mechanism (Myers and Tykhonov 2023)[7]. This framework allows customized scripts and field-specific JSON configurations to define how metadata fields interact with external services and vocabularies, thereby streamlining data entry and enhancing metadata quality.

Integration of ORCID and ROR for Researcher and Institution Identification: As part of this functionality, *DaRUS* connects with ORCID and ROR databases to enable an autofill feature for relevant metadata fields. ORCID (Open Researcher and Contributor ID) provides unique identifiers for researchers, ensuring precise attribution of research outputs and affiliations. Similarly, the Research Organization Registry (ROR) assigns persistent identifiers to research institutions, standardizing institutional affiliations in research metadata. Through this integration, users can search for an author's name to automatically retrieve and populate the corresponding ORCID ID, while searching for a university or organization suggests and autofills the appropriate ROR ID (cf. Figure 5). This feature enhances the accuracy and consistency of metadata related to researchers and institutions.

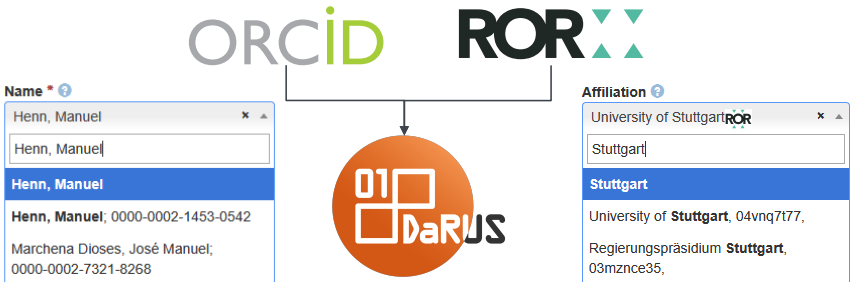


Figure 5: Interface to ORCID and ROR.

Topic Classification Using the TIB Terminology Service: To further standardize metadata, *DaRUS* incorporates topic classification based on the DFG Subject Classi-

7 https://github.com/gdcc/dataverse-external-vocab-support; *Visited on March 30, 2025.*

fication Ontology (2024–2028). Unlike broad subject classification (e.g., "Physics" vs. "Quantum Optics") or free-text keywords, which lack standardization, topic classification uses predefined, structured categories that enhance metadata consistency, discoverability, and interoperability across research repositories.

To implement this, *DaRUS* integrates the TIB Terminology Service[8], a REST API maintained by the German National Library of Science and Technology (TIB) that provides authoritative classifications and controlled vocabularies. Through this integration, users can search for a research topic, and the system suggests appropriate classification terms from the DFG Subject Classification Ontology. The controlled vocabulary name and the corresponding URI are then automatically populated in the metadata fields as depicted in Figure 6, ensuring consistency and standardization in subject classification.

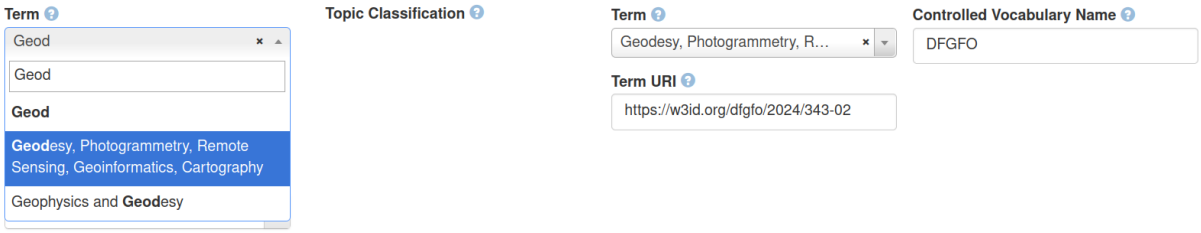


Figure 6: Interface to the terminology service to check for topic classification.

Standardizing Metadata for Project and Funding Information: Building upon these enhancements, metadata related to project details and associated funding information has also been standardized within *DaRUS*. A major challenge was the structural limitation of Dataverse, which only supports two hierarchical levels in its metadata framework. This constraint required an alternative approach to linking project and funding metadata across multiple fields simultaneously.

To address this, a functionality that allows users to search by project name or acronym has been implemented, which then autofills relevant details across multiple metadata fields. Specifically, when a project is selected, *DaRUS*:

- Fills the acronym or name in the Project metadata field.
- Simultaneously populates the Funding Information metadata field with details such as the funding agency, related project acronym, and project identifier.

This functionality is enabled through integration with the Research Information System (FIS) of the University of Stuttgart, a database-driven platform that collects, analyzes, and manages metadata on research activities, including researchers, publications, projects, and funding. By leveraging the FIS REST API, *DaRUS* retrieves structured metadata, ensuring accurate and consistent project and funding information.

8 https://terminology.tib.eu/ts; *Visited on March 31, 2025.*

A significant challenge in this implementation was handling complex scenarios where a dataset is linked to multiple projects, some with multiple funding sources. Our solution addresses this by using the project acronym as a linking key, enabling accurate autofill even in cases involving multiple grants per project. This approach ensures the structured representation of funding relationships, enhancing metadata completeness and accuracy as can be seen in Figure 7.

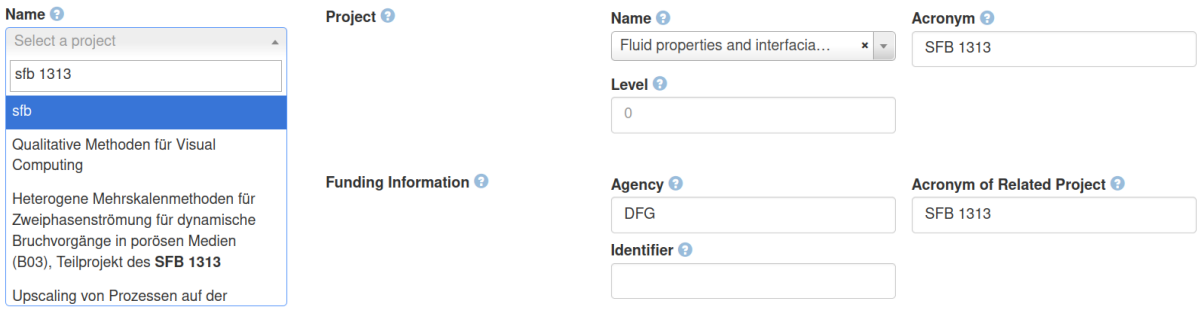


Figure 7: Interface to the research information system.

By implementing these automated interfaces, *DaRUS* significantly improves metadata standardization, reduces manual effort, and enhances the quality and interoperability of research data across repositories.

2.2 Formal Curation Support by pubWorkflow

Dataverse allows to integrate an external tool into the publication workflow as schematically shown in Figure 8(a). In our case the external tool is *pubWorkflow*, a Python REST API which can perform different tasks in the workflow. When a user clicks on "Publish" on a dataset page, the item will not be published instantaneously, but the publication will be stopped and *pubWorkflow* will be called to enable the curation workflow. On submission to our curation workflow, the dataset will be locked by default and can be unlocked by *pubWorkflow* to adapt the dataset for final publication. During the curation workflow, automatic checks will be triggered against various public APIs to facilitate the review. Table 1 gives an overview of the *DaRUS* metadata fields with validation logic implemented in *pubWorkflow*.

Existing metadata of a dataset will be sent to specific endpoints to get review recommendations and validations of a dataset's metadata.

Table 1: Metadata fields automatically checked by *pubWorkflow*.

metadata property	**DaRUS property name**	**example value**
author	authorName	Fritze, Florian
	authorIdentifier	https://orcid.org/0000-0002-9949-3815
grant_number	grantNumberValue	458524799
	grantNumberAgency	DFG
keyword	keywordTermURI	http://www.wikidata.org/entity/Q831774
	keywordValue	Density Matrix
ds_description	dsDescription	A linked README.md file in the description as HTML markup
publication	publicationIDNumber	https://doi.org/10.1002/ejic.202200709
topic_classification	topicClassValue	Optics, Quantum Optics and Physics of Atoms, Molecules and Plasmas
	topicClassVocabURI	https://w3id.org/dfgfo/2024/323

As shown in Table 2, we have integrated the ORCID API, Wikidata API, TIB Terminology Service API, Crossref API, Unpaywall API, DataCite API, Library of Congress Subject Headings API, OpenAire API and Loterre Chemistry Vocabulary API endpoints.

Table 2: API endpoints used for the automatic checks.

metadata property	**API endpoint**
author	https://pub.orcid.org/v3.0/expanded-search
grant_number	https://api.openaire.eu/search/projects
keyword	https://query.wikidata.org/sparql https://api.terminology.tib.eu/api/search https://sparql-endpoint.loterre.fr/LoterreVocabularies/sparql https://id.loc.gov/authorities/subjects/suggest
publication	https://api.crossref.org/works/ https://api.unpaywall.org/v2/ https://api.datacite.org/dois/
topic_classification	https://api.terminology.tib.eu/api/search

For instance, if there is a dataset which has a keyword with no Term URI the specific keyword will be sent to the aforementioned terminology services to automatically retrieve suggestions for Term URIs. Figure 8(b) shows an example for the keyword *Physics*. The reviewer of the dataset does not need to enter the keyword manually on various

terminology websites. If all validation checks are done, *pubWorkflow* will send a review mail to the ticket system so that the reviewers can see all the validation results. The next step will be to temporarily unlock the dataset in order to improve the metadata quality of the dataset and curate it as a *FAIR* dataset. If there is missing metadata, *pubWorkflow* will suggest it during the review. If there is already metadata in a specific metadata field, *pubWorkflow* will check if the field is matching our review standards. Let's say there is term value and a term URI for the topic classification section. PubWorkflow will validate the metadata. Another use case is the ORCID validation. If an author has no ORCID entry, *pubWorkflow* will send the author's name to the ORCID API to retrieve possible ORCID identifiers or the second case if there is an ORCID already, *pubWorkflow* will check if the author's name and ORCID identifier match with the entries in the ORCID database. So there will be suggestions for additional metadata entries and there will be also validations of current already entered metadata entries. These automated retrievals and checks can bring more efficiency to the publication workflow because there is no need anymore to do it manually piece by piece.

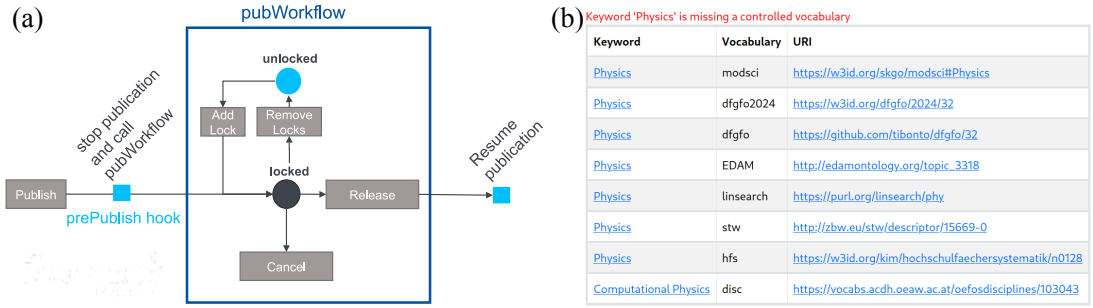


Figure 8: (a) Embedding of *pubWorkflow* in the publication workflow of Dataverse; (b) Term URI suggestion for given keyword.

3 Outlook

3.1 Scientific Curation Support

So far, the scientific review and the formal review are two separate steps in the publication workflow. To speed up the process and to allow the authors to get feedback from both scientific and formal viewpoint together, integration of *pubWorkflow* with *easyReview*[9] is planned, an online tool that allows to add comments to metadata and files and to interact with the authors in a chat-like interface.

Furthermore, an integration with the new overlay journal *JodaKISS*[10] is intended, which offers a scientific peer review for already published simulation data and software. Partic-

9 https://github.com/JR-1991/easyReview; *Visited on March 30, 2025.*
10 https://jodakiss.episciences.org/; *Visited on March 30, 2025.*

ularly high-quality data and software sets that are of distinct value to science can thus achieve additional visibility.

3.2 Automated Subject Indexing

To further improve the quality of metadata while also simplifying the process of entering said metadata for researchers, our ongoing project DA-FDM (digital assistant for research data management) focuses on enhancing the visibility and discoverability of research data within *DaRUS* through automated subject indexing. To this end, we are currently developing a tool called DA-FDM which can suggest terms from controlled vocabularies for the topic classification field in *DaRUS*. Whenever a researcher enters their research data in *DaRUS* and provides a related publication of the dataset or at least an abstract, this information will be passed to an AI model. The process then involves generating vector embeddings of the respective text, be it the related publication or the abstract, computing similarity measures against curated reference corpora, and inferring a subject classification based on nearest-neighbour relationships. The researcher will then be provided with a list of suggested terms from a controlled classification system, from which they can simply pick the terms best describing their dataset and confirm them. Thus, they do not have to think of keywords or know any subject indexing standards themselves, yet their data will be properly indexed and thereby easily findable and reusable. The information about their choices will also act as a feedback loop through which the model can be constantly improved to yield more refined results.

Acknowledgements

The development of the interfaces was partly funded by the Ministry of Science, Research and the Arts of the State of Baden-Württemberg (MWK) as part of the project "Digital Assistant for Research Data Management (DA-FDM)". The automation of the publication process was funded by the University of Stuttgart as part of the project ScalableRDM.

In addition to the authors, Anett Seeland (https://orcid.org/0000-0001-7979-8083) was involved in automating the data curation process conceptually and helped with implementing the interfaces and tools.

easyReview was written by Jan Range (https://orcid.org/0000-0001-6478-1051), the integration with JodaKISS is planned together with Sibylle Hermann (https://orcid.org/0000-0001-9239-8789).

Authorship Contributions

The presented manuscript results from joint efforts by the listed authors. In particular, section 2.2 (*Formal Curation Support by pubWorkflow*) has been written by Florian Fritze, section 2.1 (*User Input Support by Interfaces*) by Sarbani Roy, sections 1 (*DaRUS – The Data Repository of the University of Stuttgart*)and 1.1(*The publication process*) by Björn Selent and section 3.2 (*Automated Subject Indexing*) by Karoline Weinspach. All other text has been authored by Dorothea Iglezakis.

Conflict of Interest

The authors declare no conflicts of interest.

Bibliography

Iglezakis, Dorothea. 2023. *CodeMeta Metadata Block Configuration for Dataverse.* Software. Visited on March 5, 2025. https://doi.org/10.18419/darus-3291.

Jones, Matthew B., Carl Boettjiger, Abby Cabunoc Mayes, Arfon Smith, Peter Slaughter, Kyle Niemeyer, Yolanda Gil Gil, et al. 2017. "CodeMeta: an exchange schema for software metadata. Version 2.0." Edited by KNB Data Repository, https://doi.org/10.5063/schema/codemeta-2.0.

Lauterbach, Simone, Hannah Dienhart, Jan P. Range, Stephan Malzacher, Jan-Dirk Spöring, Dörte Rother, Maria Filipa Pinto, et al. 2023. "EnzymeML: seamless data flow and modeling of enzymatic data". *Nature Methods* 20 (3): 400–402. https://doi.org/10.1038/s41592-022-01763-1.

Myers, James D., and Vyacheslav Tykhonov. 2023. *A Plug-in Approach to Controlled Vocabulary Support in Dataverse.* Zenodo. https://doi.org/10.5281/zenodo.8133723.

Schembera, Björn, and Dorothea Iglezakis. 2020. "EngMeta – Metadata for Computational Engineering". *International Journal of Metadata, Semantics and Ontologies* 14 (1): 26–38. https://doi.org/10.1504/IJMSO.2020.107792.

Seeland, Anett. 2020. *EngMeta Metadata Block Configuration for Dataverse.* Software. Visited on March 5, 2025. https://doi.org/10.18419/darus-508.

Wilkinson, Mark D., Michel Dumontier, IJsbrand Jan Aalbersbergand, Gabrielle Appleton, Myles Axtonand, Arie Baakand, Niklas Blombergand, et al. 2016. “The FAIR Guiding Principles for scientific data management and stewardship”. *Scientific data* 3 (1): 1–9. https://doi.org/10.1038/sdata.2016.18.