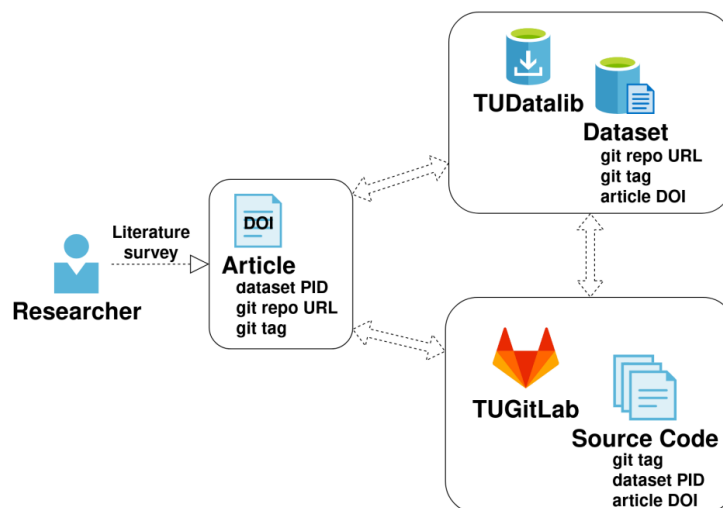


CRC 1194 guideline on linking research data with publications

Dr.-Ing. Tomislav Maric, Prof. Dr. rer. nat. Dieter Bothe, Prof. Dr. Christian Bischof

Goal

The goal of the guideline is to support **FAIR** principles in research [1] with concrete steps that each researcher can perform to increase the **F**indability, **A**ccessibility, **I**nteroperability and **R**eproducibility of their research. These steps lead to cross-linked publication (report), data that the publication is based on, as well as the source code of the software (numerical code or scripts) that were used to generate or process the data, as illustrated here:



Although alternatives exist, in the CRC 1194 at TU Darmstadt, we employ the services and software mentioned in the workflow outlined below.

Workflow

Use the git version control system [2] for your software projects and publications.

Use the TUGitlab web interface [3] to git for collaborative work.

If satisfactory results are achieved that lead to a submission of a publication

- Integrate the changes you made in your git repository of your software project into the master branch.
- It is advisable that an experienced team member checks the code quality and that the tests are running before accepting the change in the master branch.

Create a Persistent Identifier (PID) for the version of the code used to generate the results in the form of a “git tag” [4].

Clean up and publish your data on TUDatalib:

1. Document your data with a README file: information about the format used for the raw data and for the diagrams in the publication, information about the directory

structure.

2. Aim at using established Open Source formats.
3. If you stored your data in a binary format, you should also archive the version of the software used to process the data, if possible.
4. Separate the data into at least two archives, one for secondary data and another for primary data.
 - Secondary data: arrays of the curves in the diagrams, tabular data.
 - Primary data: your complete simulation or experimental results with their input data.
5. Upload the secondary data to TUDatalib and obtain a PID.
 - Add the git repo URL for the source code and the git tag to metadata.
 - Publish the secondary data and obtain a DOI: since the article is published and secondary data is a part of the article, it should be available to the general public.
6. Upload the primary data to TUDatalib and obtain a PID.
 - Ask your supervisor if primary data is to be made private or public on TUDatalib.
 - Add the git repo URL for the source code and the git tag to metadata.
7. Create an archive of your code (source and binaries!) and upload it to TUDatalib, and obtain a PID
 - Ask your supervisor if the source code archive is to be made private or public.
8. Cite the PIDs of your data and the source code into the publication.
 - Use a reference system to add references to PIDs of the code, secondary and primary data to the publication / report.

Note: The private data or code is at this point not accessible to the public, only the PIDs are visible!
9. Upload your publication to TUPrints or arXiv and obtain a PID for the publication.
 - Update the metadata of the code, secondary and primary archives with the publication PID on TUDatalib.
 - Update the git tag description in the repository with the PID of the publication.
10. Once the publication is accepted
 - Update the metadata of the code, secondary and primary archives with the publication DOI on TUDatalib.
 - Update the git tag description in the repository with the DOI of the publication.

References

- [1] <https://www.go-fair.org/fair-principles/>
- [2] <https://git-scm.com>
- [3] <https://git.rwth-aachen.de/>
- [4] <https://git-scm.com/book/en/v2/Git-Basics-Tagging>