Mélanie Fournier, 2024/09/15, FAIR seminar report.

1 Introduction

In this report, I will give an overview of the FAIR data principles. In a world where more and more data is produced, it becomes critical to define guidelines to handle that data and to make it as useful as possible for research.

The FAIR principles are one such set of guideline. FAIR is the acronym of Fair, Accessible, Interoperable, Reusable. While paraphrasing the principles in this report would be possible, the reader is referred to the website for a more in depth explanation of them. In summary, it states how the data should be stored, and documented, so that it may easily be used by users. The data should also have a license that states what a user is allowed to do with the data or not.

2 Software development guidelines

Software, unlike data, has its own history with regard to accessibility and licensing. It is through modifying and reusing code that the current software products are built. But while the standard for software engineering are quite mature at this point, there are some particularities for scientific software and data that are interesting to note.

The paper "Guidelines for collaborative development of sustainable data treatment software" approach software engineering through the scientific computing applications.

These guidelines take the FAIR standard into account, applied to software.

They can be summarized into multiple categories, which naturally overlap.

- Making code easy to develop in group: Use version control such as GIT. Assign roles that have executive power over the development. Make use of code reviews before adding some code to the software. Release often.
- Making code maintainable: Refactoring should be done with tests to ensure that nothing breaks done in the code. Large scale refactoring should be properly planned before being made, as this has the potential of being very time-consuming with no guarantee that the new architecture will work. Use CI/CD. Write some automatic tests.
- Making code convenient for the end user: Choose a user interface, being either a GUI, CLI, API. . . . Documentation with numerous examples is also primordial to produce.
- Making the code output as transparent as possible. The code output should be enriched with metadata so that the output can be used downstream. Provide user with ways to export plot as vector graphics.
- Making code readable: Use automatic formatting tools. Write code in English. Avoid optimizations that would be caught by the compiler.

There are other recommendation which are in the more in the realm of producing good quality code. These recommendations are good practices that apply to any software engineering application, such as for example making tasteful use of design pattern, making sure errors are handled well and so forth.

The bullet points above can be reflected in the FAIR principles.

Version control system allows the code and its release history to be sufficiently documented and make it accessible easily by posting for example the repositories.

Some guidelines are also aimed to make the software as accessible as possible for the end user, such as writing example.

The authors of the paper insists on the particularity that scientific code should be stable as the measuring instruments providing the data that the pieces of software use can have lifespan of decades. This is reflected in the guidelines to make sure the code is interoperable, through rigorous automated testings. The recommendation to use English also plays a role in this as this is the most widely understood language.

Finally, the guideline to write good code makes it reusable.

All of these guidelines show that while code is a special type of data, it can be treated in a way that makes it compliant with the FAIR principles.

3 Small addition on Large Scale Research Facilities

I recently had the pleasure of going to the MAX IV synchrotron to participate in experiments, and this generated a lot of data. This is interesting because the paper authors above also comes from the same kind of background. My reflection would now be to ask if this data is compliant with the FAIR principle, which they claim to be.

Before we dive into the data itself, let's have a look at the surrounding ecosystem.

A cursory look at the website and reveals some effort for transparency. There are multiple review report, usually one for each beamline, as well as annual reports and the resources used at the facility are presented, such as the IT infrastructure.

As far as I know, running experiments at MAX IV comes with an obligation to publish. The publications are available and browsable through the website, and the DOI for each of them is shown. There are some public data related to the publication, although I have no idea if there is an obligation to make the data used in publications public.

Overall, I found the website quite nice to use and there is a clear commitment to open science, at least from an exterior point of view.

Now come the data. Is it FAIR? Data gathered from experiment is automatically stored on their server. In that sense, the data is findable. It is stored in the HDF5 format which is a format that is easy to read on python, and is an open format. Each scan at the beamline produces raw data, which is stored, as well as metadata which is stored in a master file. Some processing was also done with it to make it easier to use, while still keeping the raw sensors data, so there is no loss of information.

Making it accessible is another matter. As the data is stored on remote server, it becomes necessary to authenticate to work on it remotely. There are two ways of doing so, either work via a VPN or download the data to my computer. Both methods have clear documentations that were easy to follow.

I have some concerns about using the VPN software which is proprietary, and you have to download an out of date version that I could not update. File transfers support open protocols such as SFTP, but the recommended software to use is GLOBUS, which is managed by a non-profit.

Authentication is however a bit of a tossup if it will work, which ampers accessibility somewhat.

Overall, I would say that the FAIR principles are well respected. I did not look, however, on their plans to keep the data on their server in the long term.

4 Licensing

Choosing a license for data is important, as it is a way to communicate clearly to other people what they are allowed to do with it.

For a personal project, I would have no reason to disallow modifying or extending my code. I think, however, that proper attribution is a nice thing to have.

Whether to allow commercial applications or not is more difficult to answer, and is a bit of headache. On the one hand, restricting a project to non-commercial uses guarantee that my work or subsequent work would not be used without compensation to the authors. On the other hand, running a non-commercial license poses the risk that the software will not be used at all and have reduced visibility. This is not an issue for low scale project where I would not be expecting anyone to really use my work, but for larger scale projects, this is primordial to keep the momentum and development going.

In either case, I think the data/software or any modification of it should be openly accessible. There are simply too many risks associated with doing the opposite.

Based on this and the license picker, a CC BY-SA license would work, or a GPL-3.0 license.

https://www.maxiv.lu.se/ https://ufal.github.io/public-license-selector/ https://creativecommons.org/licenses/by-sa/4.0/ https://opensource.org/license/GPL-3.0 https://content.iospress.com/articles/journal-of-neutron-research/jnr220002ref242

5 Analysis of an old paper