Codefair - Your Personal Assistant for Developing FAIR Software

Dorian Portillo, BS[1]; Sanjay Soundarajan, MS[1]; Jacob Clark, BS[1]; Bhavesh Patel, PhD[1]

[1]FAIR Data Innovations Hub, California Medical Innovations Institute, San Diego, CA, USA

Background

Making research software reusable is essential for enabling reproducible and transparent research, preventing duplicate efforts, and ultimately increasing the pace of scientific discoveries.

The Findable, Accessible, Interoperable, Reusable Principles for Research Software (FAIR4RS Principles), provide high level instructions for optimizing the reusability of research software. Following these principles requires, for instance, to:

- Specify a clear usage license
- Follow best coding practices (e.g. PEP 8 for Python code)
- Document software (e.g., in a README)
- Include metadata files such as CITATION.cff and codemeta.json
- Archive software (e.g., on Zenodo)

Problem

Researchers lack awareness, training, and time for fulfilling most of these steps. For instance, in our experience, updating the metadata files (authors, version #, release date, DOI, etc.) before each release is the most time consuming and prone to errors.

Solution: Codefair

We are developing Codefair, the first free and open source application that acts as a personal assistant for making any research software FAIR.

By leveraging tools such as Probot and the GitHub API, Codefair monitors activities on the software repository and communicates with the developers via a GitHub issue dashboard that lists issues related to FAIR-compliance. For each issue, there is a link that takes the developer to the Codefair user interface (UI, built with Nuxt 3 and Typescript) where they can better understand the issue, provide inputs to address it through an intuitive interface, and trigger a pull request to fix the issue.

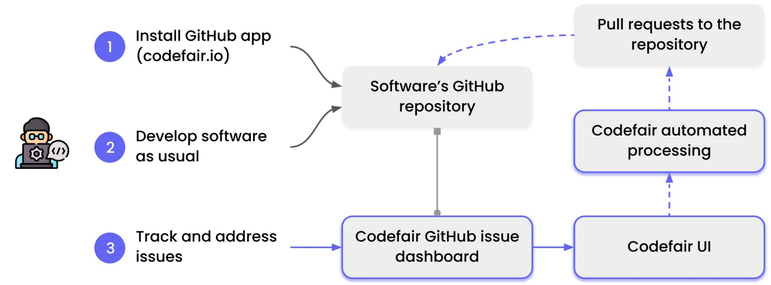

Figure 1. Overall user workflow of Codefair. After installing the Codefair GitHub app on a repository, users can develop their software as usual. They can track and address any FAIR-compliance issues easily through the Codefair GitHub issue dashboard and Codefair UI.

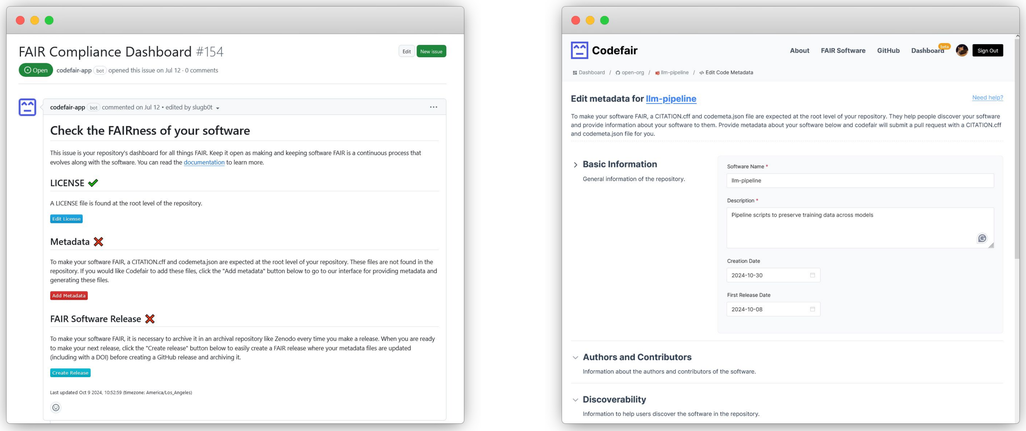

Figure 2. Left: Screenshot of the Codefair GitHub Issue dashboard. Right: Screenshot of the Codefair user interface for providing metadata and adding CITATION.cff and codemeta.json metadata files to a GitHub repository.

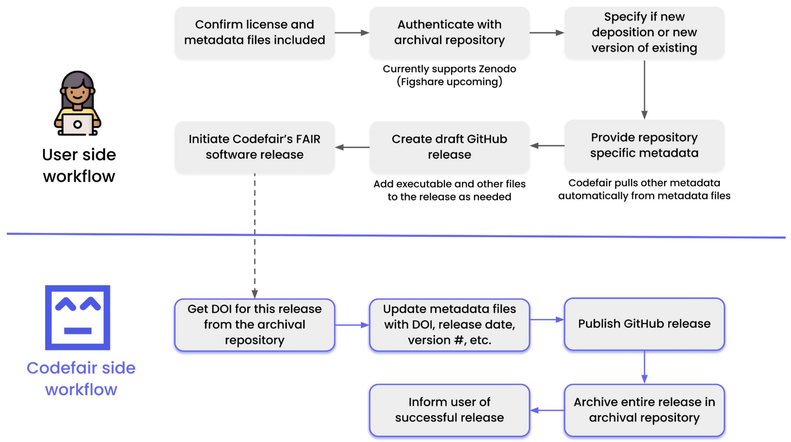

Figure 3. Workflow of the FAIR software release with Codefair. It addresses a major issue with the current Zenodo-GitHub Webhook method that doesn't provide the DOI until the software is archived on Zenodo, which prevents from including the DOI in the software metadata.

Results

Codefair is designed to be intuitive and easy to use. It integrates seamlessly into the typical software development process through GitHub (Fig. 1). It doesn't require any knowledge of FAIR or specific coding skills (Fig. 2). Automation, such as pre-populating metadata and getting license terms, is included to support users further.

While Codefair is based on the FAIR Biomedical Research Software (FAIR-BioRS) Guidelines, most elements of the guidelines are applicable to all software. Therefore, Codefair can be used by anyone developing research software in any field of research.

The current version of Codefair (v3.0.0) includes support for adding a LICENSE file, creating CITATION.cff and codemeta.json metadata files, validating Common Workflow Language (CWL) files, and making FAIR software releases through Zenodo (Fig. 3).

Try Codefair today at codefair.io. We welcome external contributions. Instructions and tutorials are provided in the Codefair documentation at docs.codefair.io.

Discussion

Additional features are being added to provide support for complying with best coding practices, archiving on Software Heritage, and much more to cover all the requirements for making software FAIR.

The usage of Large Language Models (LLMs) is also anticipated to further support users in preparing documentation, metadata, and more. This will be possible through a recent support from the Microsoft AI for Good Lab providing Azure credit to the project.

We believe Codefair is an essential tool for enabling software curation at scale and turning FAIR software into reality. The impact of Codefair can be very broad as it can also be extended to support aspects outside of FAIR, such as software quality and security.

To expand the FAIR-BioRS guidelines to all fields of research, we have started a new Task Force through the Research Software Alliance (ReSA). Join the task force and contribute to developing actionable guidelines for making software FAIR:

www.researchsoft.org/taskforces

Bhavesh Patel
bpatel@calmi2.org
fairdataihub.org

Find this poster and all related references here

bit.ly/codefairUSRSE2024