Silicon Valley Community Foundation - Essential Open Source Software for Science Cycle 5 Progress Report

Grant Overview

Grant title: Improving OpenRefine's reproducibility

Grant number: EOSS5-0000000404

Date report submitted: 2024-08-21

Lead applicant name: Antonin Delpeuch, Martin Magdinier

List of key personnel directly funded by this EOSS award (including the lead)

Name	Email	Open source project affiliation
Antonin Delpeuch	antonin@delpeuch.eu	OpenRefine
Freexian	raphael@freexian.com	Debian
Zoe Cooper	zoe@zoe-cooper.com	OpenRefine

Financial Overview

Provide original estimated and actual spending for each category for the reporting period. You can also submit this information as a separate spreadsheet.

	Original Yearly Budget	Original Total Budget	Year 1 Actual	Year 2 Actual	Total Actual	Difference
(1) Development	$63,000.00	$126,000.00	$55,222.82	$41,520.00	$96,742.82	$29,257.18
(2) Design and design research	$8,000.00	$16,000.00		$38,400.00	$38,400.00	-$22,400.00
(3) Training and documentation	$10,000.00	$20,000.00			$0.00	$20,000.00
(4) Product management	$28,800.00	$57,600.00			$0.00	$57,600.00
(5) Community outreach	$10,000.00	$20,000.00			$0.00	$20,000.00
(6) Travel meeting equipement and PR	$12,000.00	$24,000.00	$122.94	$5.27	$128.21	$23,871.79
(7) Personel fringe	$0.00	$0.00	$778.92		$778.92	-$778.92
Fiscal sponsorship fee (15%)	$23,250.00	$46,500.00	$23,257.50	$23,257.50	$46,515.00	-$15.00
TOTAL	**$155,050.00**	**$310,100.00**	**$79,382.18**	**$103,182.77**	**$182,564.95**	**$127,535.05**

Budget narrative: Briefly explain any changes to the original budget, any challenges impacting spending; unspent budget; and plans for use of remaining funds.

(1) As explained in our previous progress report, the development work started on January 1st, 2023, instead of November 1st, 2022. For year 2, only eight months from Antonin are spent on this grant. The other four months are invoiced on a separate grant or are on two-month unpaid leave.

(2) We combined the design and training/documentation budget into a larger package. This helps us hire a designer for a longer period (6 months) instead of working through mini contracts.

(3) The training/documentation budget was merged with the design budget.

(4) Through the grant the product and project management was done by Martin Magdinier paid via another grant.

(5) The designer took care of the community outreach by scheduling several interviews with diverse user groups. We did not spend that budget.

(6) Our main event in Berlin in 2024 was financed via another grant, limiting our spending on this budget through the grant.

(7) This is a new budget line introduced in our previous report. It represents funding for a bug bounty platform.

Please list any other funding received during this reporting period that <u>extends the work you are doing on this project</u> (if relevant):

Award Title	Funding Source	Award Amount (US$)

Progress Overview

Progress towards the deliverables:
Please provide a summary of milestones / deliverables achieved or significantly contributed to through this grant as a bulleted list (short description + optional notes)

The transformations made in an OpenRefine project are recorded in an undo/redo history, letting users go back to any previous state of their data cleaning process. Beyond this, the tool also offers a way to extract the list of transformations into a machine-readable format, which can then be used to re-apply them later on a new dataset.

Although this feature can readily be used to share a series of cleaning steps with fellow users to let them use it in their own projects, its reliability falls short of the expectations of a reproducible data cleaning platform. Any discrepancies between the structure of the dataset can lead to errors. Those are poorly handled, without giving the user an opportunity to adjust the recipe to the dataset it is being applied to. The format in which the recipes are represented, a JSON representation of operation parameters, is also cumbersome for users to read (making it difficult to understand what a recipe does) or to manipulate (adjusting the parameters of an operation in the recipe can render the recipe invalid).

In this project, we have been concerned with making changes to OpenRefine's operation history to address those problems. Reworking the structure of the operation history is also an opportunity to address usability issues for everyday users, beyond the strict use case of reproducing a workflow used in a research project.

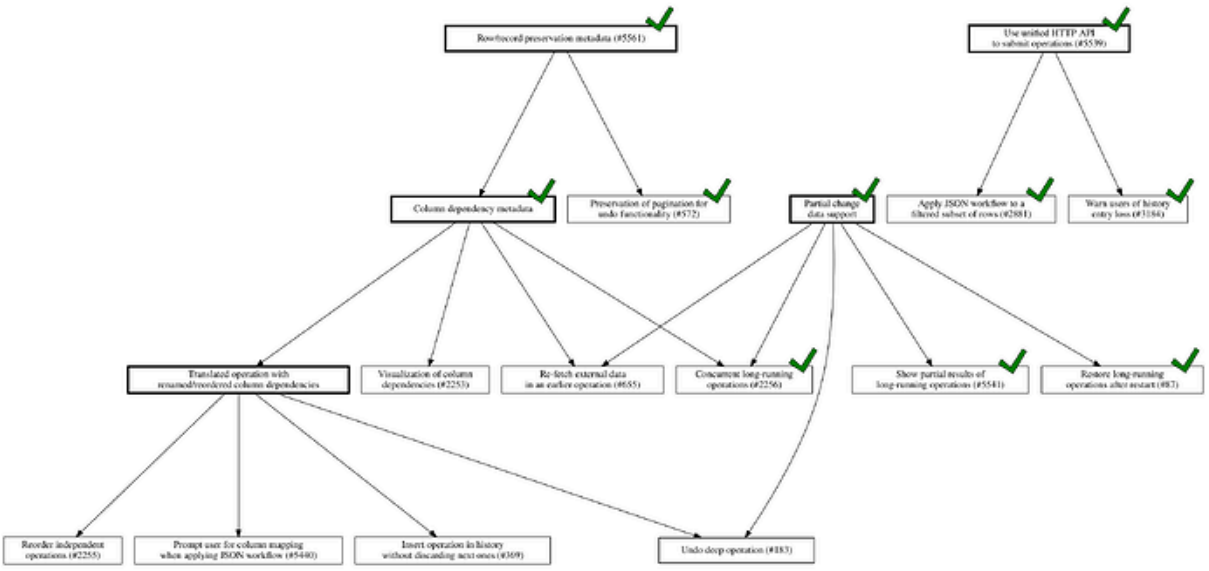
At this stage of the project, we have designs of the proposed improvements and an implementation of the underlying changes to the representation of the operation history.

The following diagram (taken from the corresponding forum thread where it can also more easily be zoomed in on) gives an overview of the work packages for which an implementation is available on the delta branch:

- Displaying results of an operation before its computation is complete: https://forum.openrefine.org/t/partial-results-of-long-running-operations/482
- Pausing and resuming operations: https://forum.openrefine.org/t/partial-results-of-long-running-operations/482/3
- Concurrency of operations: https://forum.openrefine.org/t/concurrency-of-long-running-operations/1009/8

We also have mock-ups of the proposed UI changes for the remaining improvements on the operation history:

- Additional actions available on history entries: https://forum.openrefine.org/t/four-operation-history-mockups/1362
- Introduction of operation icons to identify operations more easily: https://forum.openrefine.org/t/operation-logos/1307
- Graph-based representation of the history structure and improvements to the error handling of the application of a recipe: https://forum.openrefine.org/t/operation-history-mockups/1557



Major changes in scope or project plan:
Please describe any problems encountered and steps taken to overcome them, as well as any general scope or plan changes.

Review and release strategy

The main hurdle we have encountered is the question of how those changes should be reviewed by the community and delivered to users. The implementation strategy outlined above relies on the ability to make sweeping changes throughout the code base, including breaking architectural changes. Given

that we want to minimize breaking changes to keep the adaptation burden on extension maintainers as light as possible, we have been postponing the release of those changes to try and validate ahead of time that they indeed enable the new features they are introduced for. The downside of this approach is that it has led to a large backlog of unreleased functionality, which is difficult to review for the developer community.

Strategy for extensions
The delivery of such changes to the user community is made more difficult by the fact that the user experience around updating OpenRefine is rather poor when the user has installed extensions. Extensions can easily break when OpenRefine is updated, primarily due to an insufficient isolation of core and extension code, unclarity around officially supported extension points and their intended stability and complicated extension installation and update procedures. There are ongoing discussions in the community as to how best to address this.

Designer onboarding
By hiring a designer for this project, our aim was to onboard someone to work on this specific reproducibility improvements project, but whose role would then grow into a more general Lead Designer role, participating in improvements in other areas of the product. We wanted to bring onboard someone who would be able to relate to our users' needs and struggles (including new users) and therefore did not prioritize existing knowledge of the tool in our hiring.

While this partly succeeded, for instance to identify usability issues from the perspective of an untrained eye, the onboarding effort was significant. The designer had to not only understand how the tool is used but also get used to the organization of work in an open source project like ours, where the team is very small no other designers are currently active. Overall, our support was not sufficient for the designer to become autonomous in the project.

Proposed new approach
Instead of basing our reproducibility improvements on large architectural changes, we have started working on implementing them as lightweight changes on top of the existing architecture. This will prevent us from addressing certain systemic issues, such as validating the structure of a list of operations before applying it on a dataset, or the ability to make changes to older operations without discarding the subsequent ones. But we should still be able to deliver most other changes, such as an improved user experience around extracting and re-applying operations and better visualization of the history.

The approach we propose is as follows:

- Add metadata to existing operations, in a backwards compatible way, so that they expose their expectations on the table structure (input columns, overlay models required) as well as the expected impact they have on the grid (columns created or deleted, preservation of record boundaries). This metadata is speculative, in the sense that the architecture does not validate that the operation fully respects it, but it enables a better error reporting ahead of applying a series of operations, but it should be sufficient to enable the following user-facing improvements
- Implement a better visualization of the operation history, following the existing designs. The metadata exposed earlier can be used to show the operation dependencies and the operation icons can help understand the structure of the history in a more visual way
- Implement better error-reporting and validation to apply recipes, also based on the metadata introduced earlier.

We believe that with this approach, we will be able to deliver the majority of our planned changes to users, on OpenRefine 3.x, without them being dependent on the release of breaking changes and the coordination they require.

CZI Community Building: *Please describe any ways in which you feel you and/or your team benefited*

from general CZI and/or open science community building activities (e.g., convenings, webinars, trainings, CZI's Movement and Capacity Building offerings, etc.). If you don't feel you benefited, please describe what CZI could change or offer that would be more valuable to you.

We are monitoring CZI Capacity Building Opportunities and two members of the project attended the CZI Open Science conference held in Boston in June. We valued the conference a lot to build relationships with other projects.

Diversity, Equity and Inclusion

Briefly describe the activities undertaken during this reporting period to **increase the diversity of contributors, leaders, and users, and/or to improve the inclusivity of the project(s)**.

Most of the Diversity action takes place under the EOSS-Diversity grant which was extended until December 2024. We are continuing our participation in internship programs with Google Summer of Code and we organized our first in-person contributor meeting in Berlin in June 2024.

Key Deliverables and Project Recognition

If new preprints, journal or conference publications, methods/protocols, datasets, documentation, new code repositories were created or deposited through this grant, please provide identifiers/URLs.

Identifier (URL or DOI)	**Title**	**Notes** (optional)

Additional recognition for the open source project(s) and/or key personnel:

Collaboration

Have you collaborated with, benefited from, or used resources produced by other CZI associated projects or researchers (e.g., other patient or research organizations, community partners, etc.) outside of your immediate funded project team? If so, please complete the table below:

Name	**Organization**	**Description of resource exchange or collaboration**	**Was this collaboration started, renewed, or continued as a result of a CZI-sponsored event/ activity?** Please respond yes/no, and if yes, which event or activity.
