NGI Grant Proposal: OrbitMines' (Hypergraphic) Version Control System through Rays

A grant proposal to NGI through NLnet. In contrast to my previous application, which I realized was quite naive after a conversation with Michiel Leenaars at NGI FORUM 2023: Hopefully this will serve as a concrete target problem to solve.

OrbitMines Research

Fadi Shawki
fadi.shawki@orbitmines.com
FadiShawki · @_FadiShawki · fadishawki · @fadishawki
0009-0009-9288-992X

31 January 2024

Please select a call

Thematic call

In the list of current calls below, please indicate the call topic you are responding to. Note that some larger funds (like NGI0 Core and NGI0 Commons Fund) (part of the Next Generation Internet initiative) will have some special scope or conditions. You'd better have a look at them before you submit a proposal. If in doubt, submit to our open call and mention in the application that you are okay with us allocating your proposal to the most suitable fund.

Open Call

Contact information

Your name (max. 100 characters)

Fadi Shawki

Email address

fadi.shawki@orbitmines.com

Phone number

+31 6 84704186

Organisation (max. 100 characters)

OrbitMines

Country

The Netherlands

General project information

Project name (max. 100 characters)

OrbitMines' (Hypergraphic) Version Control System through Rays

Website / wiki

orbitmines.com

Abstract: Can you explain the whole project and its expected outcome(s).

Please be short and to the point in your answers; focus primarily on the what and how, not so much on the why. Add longer descriptions as attachments (see below). If English isn't your first language, don't worry — our reviewers don't care about spelling errors, only about great ideas. We apologise for the inconvenience of having to submit in English. On the up side, you can be as technical as you need to be (but you don't have to). Do stay concrete. Use plain text in your reply only, if you need any HTML to make your point please include this as attachment. (you have 1200 characters)

A quick description of this Version Control System is as follows: Through a self-similar reformulation of arbitrary Hypergraphs I've called Rays. Be able to restate the problem of casual histories, as follows:

You (the user) gives the library your data structure you're interested in preserving. Any types, numbers, mutability of them in the form of possible functions to apply to them, can all be optionally passed to this library (In the formulation of Rays, this just means traversal: Everything is traversal in this data structure).

The system will compile or provide at runtime the histories desired of your data structures. Imperfect recall will be supported. With possibilities of only showing partial histories or supply a way of inferring them from surrounding structure. Arbitrary step-wise traversal will be supported.

A practical problem here, being the search problem to find more optimized (or performance-heavy) variants for the particular data structure the user is interested in.

So in short terms. Better infrastructure for CRDTs and Version Control Systems and abstract them away as much as possible.

Have you been involved with projects or organisations relevant to this project before? And if so, can you tell us a bit about your contributions?

(Optional) This can help us determine if you are the right person to undertake this effort

Requested support

Requested Amount

(between 5000 and 50000)

50.000 €

Explain what the requested budget will be used for?

Does the project have other funding sources, both past and present? (If you want, you can in addition attach a budget at the bottom of the form). Explain costs for hardware, human labor (including rates used), travel cost to technical meetings, etc.

There are no other funding sources other than my own in sustaining myself while doing research the last couple of years.

- Hardware requirements: A tinybox or an equivalent in case of import-related issues. 15.000 USD excl VAT. Corrected for EUR, taxes, shipping costs and a small buffer in case of import issues: 17.500 EUR should suffice.

- 12.500 EUR to sustain myself for at least a year (any additional costs associated with the project will come out of this).

- 20.000 EUR to involve a second person in the project. (I might do something more non-trivial, but a second person is the simplest case)

If the deciding factor for the grant is any of these. They can be relaxed according to what NLnet/NGI deems appropriate.

Compare your own project with existing or historical efforts.

(e.g. what is new, more thorough or otherwise different)

A full (and live) comparison can be found here: Project - (Hypergraphic) Version Control System through Rays (2024). The short summary being that this project is the generalization of many previous approaches and focuses on a proper programmatic infrastructure (This includes CRDTs, ...).

What are significant technical challenges you expect to solve during the project, if any?)

(optional but recommended)

- Arbitrary (partial, ..., incomplete) history of arbitrary functions, rewrites, ..., data types (generalized to some arbitrary Ray).

- This then includes the automatic inclusion of existing data structures

- Automatically includes DPO(I), ..., Git histories.

- Allow for the realization that any arbitrary part of the rays are destroyed (this could include local connectivity).

- Can have a history, but no current value. No structure, ..., type, permutation information on, ..., between vertices except for that of something we identify as history.

- Arbitrary initial setup not necessarily optimal version. And through time find more optimal one's given a particular situation. (Hence just apply the same version control system on the version control system itself.)

- Thus requires the automatic shifting between possible ways of representing histories.

- This thus includes a way of mapping perceived geodesics.

- Unrolling of loops, ..., orbits, and setup argument for what kind of information one needs to differentiate between iterations of that loop, and possibly what certain initial places were.

Describe the ecosystem of the project, and how you will engage with relevant actors and promote the outcomes?

(E.g. which actors will you involve? Who should run or deploy your solution to make it a success?)

In essence this shouldn't be more than a single properly crafted file. It will be able to compile (or provide compilation) to any setting someone would want to use it in. As infrastructure, probably any project will be able to use it - the difference in use-cases will merely be to what extend histories are preferred to be kept.

Footnotes & References