Fun with VCS - more with less

A Tool for Facilitating the Use of Git in Linguistic Research Data Management

Ferger, Anne
anne.ferger[at]uni-paderborn.de
Universität Paderborn, Deutschland
ORCID-iD: 0000-0002-1382-2658

Jettka, Daniel
daniel.jettka[at]uni-paderborn.de
Universität Paderborn, Deutschland
ORCID-iD: 0000-0002-2375-2227

Introduction. The work with digital data demands some form of version control. In its simplest and still common appearance version control consists of file naming conventions, and often some creative form of duplication of files. The rising degree of collaboration in data creation and complexity of data, however, brings these basic methods to their limits, which is one of many practical and theoretical reasons why explicit and structured version control with the help of a version control system (VCS) is required. The introduction and use of a VCS for research data in the humanities faces the obstacle of a steep learning curve and poses severe problems with regard to usability, since VCS implementations mostly originate from software engineering. To deal with this problem we discuss the open source script LAMA which is used to perform operations with the VCS Git. Using LAMA does not require in-depth knowledge of the VCS or additional technical expertise, so the focus of the individual researcher can remain on the creation of the research data itself. Additional advantages include messenger integration and the option to run consistency operations with pluggable tools

1 Introduction

Using a version control system (VCS) for organizing and collaborating on research data yields multiple benefits for the quality control, reusability and reproducibility of the created resources. With VCSs originally designed for software development, the acquisition of a certain degree of technical expertise in combination with conceptual and operational specifics are often required for their application. This hampers the application of VCSs in areas which do not have a strong connection to software engineering, like the humanities.

To deal with this problem the open source script LAMA (Linguistic Automation Management Assistant)[1] was created to allow for a neat communication of inexperienced users with the VCS Git[2]. Using LAMA does not require in-depth knowledge of the VCS or additional technical expertise (it runs on Windows, Linux, and Mac), so the focus of the individual researcher can remain on the creation of the research data itself. Additional advantages include messenger integration and the option to run consistency operations with pluggable tools. While the hurdle of using the VCS is minimized for inexperienced users, its advantages with all its functionality can still be leveraged by more technically experienced collaborators.

2 Version control for research data in the humanities

Version control enhances the integrity of research data and can be seen as a prerequisite for producing replicable, high-quality research data[3] and software[4]. It is a technical means to tackle the call for replication of empirical research results in the humanities[5].

2.1 Usability and technical proficiency when employing a VCS

The handling and application of version control should become a natural skill in the digital literacy of empirically working researchers in the humanities and beyond. The transition phase, however, has to be supported by technology and workflows which foster the immediate and correct operation of VCSs and prevent support and maintenance cycles wherever possible.
Most VCSs come with a multitude of clients, often in the form of graphical user interfaces (GUIs)[6] that aim for easier handling of the version control mechanisms, i.e. enhance usability. However, they are also forced to represent the full (or at least most of the) functionality of a VCS because most clients target a broad and discipline-overarching user community.
Many factors favor the most commonly used VCS Git, but there are two main disadvantages, which are the usability and user-friendliness as

[1] Ferger/Jettka, 2021.
[2] Software Freedom Conservancy, Inc. “Git.” - The concept of LAMA could be extended to work with other VCSs as well.
[3] e.g. Klump et al., 2020. and Beckman et al., 2021 for the field of statistics
[4] e.g. Scheliga et al., 2019.
[5] cf. Peels/Bouter, 2018.
[6] e.g. Kernel, “Git Interfaces.”.

well as the handling of binary files[7]. Both play an important role in (linguistic) research data creation and management. While using Git LFS[8] (Large File Storage) can help with the handling of (large) binary files, learning Git and using one of the available GUIs is one of the main hurdles keeping researchers (in linguistics or generally in the humanities) from using Git as a VCS for their research data.

2.2 Meeting the challenges with LAMA

To overcome the usability challenges and the obstacle of a very steep learning curve, a new approach was tested to minimize the communicative problems between users (in this case linguistics researchers) and VCS. The result is the implementation of LAMA. The shell script LAMA serves as a basic Git client adapted to the needs of basic Git users, following minimalistic principles. By focusing purely on the operations needed for (linguistic) research data LAMA facilitates the use of the VCS, while at the same time minimizing the risk of data loss caused by accidental misuse. Dependencies and the visual design also concentrate on the absolute essentials, which enables wider use cases and a straightforward setup of the script.
The user interface of LAMA consists of a text-based menu (see Figure 1) with basic operations[9] sufficient for working collaboratively on (linguistic) research data.

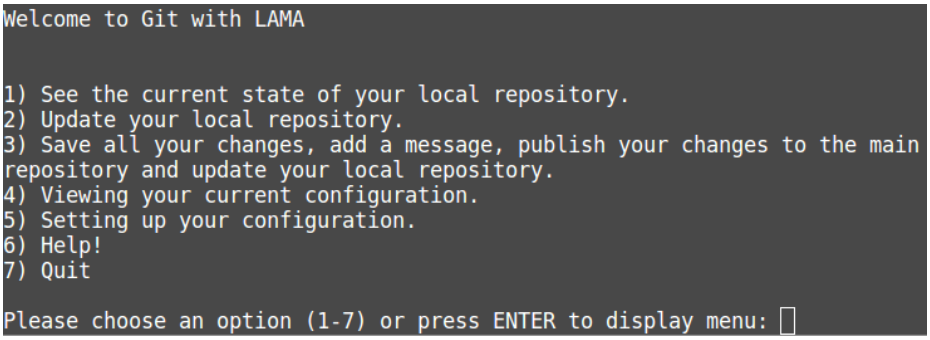

Fig. 1. Screenshot of the LAMA menu.

[7] cf. Hermann et al., 2018: 19.
[8] GitHub, Inc. "Git LFS."
[9] For administration and support purposes a "secret" option 0 is also available which allows for entering any command, e.g. more complex Git operations.

Besides its simplicity, a main advantage of LAMA is the possibility to induce operations into the version control workflow without adding complexity for the users. For instance, a slightly more complex version exists that contains integration with a messenger (tested with Mattermost[10]) to allow for reports or error messages to be sent automatically, which can potentially be used for diagnostic purposes. Furthermore, mechanisms have been successfully implemented for enhanced handling of semi-structured file formats (e.g. XML files) which often pose challenges to VCSs. In this manner, it is possible to add further operations for quality assurance, client-side data cleaning and reporting in LAMA.[11] A German version of LAMA is also available.

3 Use cases for LAMA and further work

While LAMA has already proven useful through its successful application in a research project that produces complex, high-quality linguistic data, some important questions remain. Firstly, it should be discussed if the concept can be transferred directly to other projects, and especially how target groups in the humanities (or elsewhere) can be identified and reached who do not have much contact or direct access to digital/DH experts who could assist with the implementation of reliable version control. Secondly, the actual performance and usability of LAMA would have to be thoroughly tested and compared to other VCS clients. Feedback on those questions, possibly supported by a new feedback option in LAMA, would be highly valuable for the further development.

Currently, there are limits in the complexity of version control operations that are covered. Features of Git which are used in sophisticated data creation workflows and highly collaborative contexts (e.g. branching) are not supported directly (or at least not controllable by the target user). This on the one hand is a conscious design decision (as it clearly adds to the simplicity), but on the other hand certainly poses a limit to the contexts of its application. However, for more advanced operations a multitude of other clients are available.

While in the current version of LAMA automated webhooks are only used for the messenger integration, this option can be used in a much more versatile way to facilitate the integration into any existing infra-

[10] Mattermost, Inc., “Mattermost.”.

[11] Similar quality control methods are available natively in Git, e.g. in the form of Git Hooks. Their main disadvantage is that they have to be configured individually for each user/data repository.

structure, technical setup and data creation workflow. For instance, automatic creation of issues in a project management system or in GitHub[12] could be helpful. By providing automatic issue creation in an issue tracker directly related to the LAMA code, an efficient feedback option could be implemented in the future.

4 References

Beckman, M. D., Çetinkaya-Rundel, M., Horton, N. J., Rundel, C. W., Sullivan, A. J. & Tackett, M. (2021). "Implementing Version Control With Git and GitHub as a Learning Objective in Statistics and Data Science Courses", *Journal of Statistics and Data Science Education*, 29:sup1, S132-S144. https://doi.org/10.1080/10691898.2020.1848485

Ferger, A., Jettka, D. (2021). "LAMA - your friendly and easy git script (3.0)". Zenodo. https://doi.org/10.5281/zenodo.4725651

GitHub, Inc. "GitHub." GitHub. The largest and most advanced development platform in the world. Accessed April 30, 2021. https://github.com

GitHub, Inc. "GitLFS." Git Large File Storage. An open source Git extension for versioning large files. Accessed April 30, 2021. https://git-lfs.github.com/.

Kernel. "Git Interfaces." Git. Interfaces, Frontends, and Tools. Accessed April 30, 2021. https://git.wiki.kernel.org/index.php/InterfacesFrontendsAndTools

Hermann, F., Pietsch, C., & Cimiano, P. (2021). "Conquaire Infrastructure for Continuous Quality Control". *Studies in Analytical Reproducibility: the Conquaire Project*. http://nbn-resolving.de/urn:nbn:de:0070-pub-29517575

Klump, J., Wyborn, L., Downs, R., Asmi, A., Wu, M., Ryder,G. & Martin, J. (2020). "Principles and best practices in data versioning for all data sets big and small". Version 1.1. Research Data Alliance. https://doi.org/10.15497/RDA00042

[12] GitHub, Inc. "GitHub.".

Mattermost, Inc. “Mattermost.” Mattermost. An open-source collaboration tool for developers. Accessed April 30, 2021. https://mattermost.com/

Peels, R., Bouter, L (2018). “The possibility and desirability of replication in the humanities” *Palgrave Commun* **4,** 95. https://doi.org/10.1057/s41599-018-0149-x

Scheliga, K. S., Pampel, H., Konrad, U., Fritzsch, B., Schlauch, T., Nolden, M., zu Castell, W., Finke, A., Hammitzsch, M., Bertuch, O., Denker, M. (2019) “Dealing with research software: Recommendations for best practices”, Potsdam. https://doi.org/10.2312/os.helmholtz.003

Software Freedom Conservancy, Inc. “Git.” Git. Free and open-source distributed version control system. Accessed April 30, 2021. http://git-scm.com/