Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 1 April 2019 doi:10.20944/preprints201904.0008.v1

Article

Enabling fine-grained scientific citation for GRASS GIS software modules

Peter Loewe [1]*, Vaclav Petras [2], Markus Neteler [3] and Helena Mitasova [2]

1 DIW Berlin, Mohrenstraße 58, Berlin, Germany; ploewe@diw.de
2 Center for Geospatial Analytics, North Carolina State University, Raleigh, NC, USA
3 Mundialis GmbH u. Co. KG , Kölnstrasse 99, 53111 Bonn, Germany
* Correspondence: ploewe@diw.de; Tel.: +49-30-897-89-222

Abstract: The authors introduce the GRASS GIS add-on module g.citation as an initial implementation of a fine-grained software citation c oncept. The module extends the existing citation capabilities of GRASS GIS, which until now only provide for automated citation of the software project as a whole, authored by the GRASS Development Team, without reference to individual persons. The functionalities of the new module enable individual code citation for each of the over 500 implemented functionalities, including add-on modules. Three different classes of citation output are provided in a variety human- and machine-readable formats. The implications of this reference implementation of scientific software citation for both for the GRASS GIS project and the OSGeo foundation are outlined.

Keywords: GRASS GIS; g.citation; software citation; open science; OSGeo; credit; rewards

---

Introduction of GRASS GIS 30-year-long software development

GRASS GIS [1] is a community-driven software project already lasting over three decades with continuous community-driven development and maintenance efforts. Since 1983, the software has continuously evolved and its capabilities have been continuously extended according to the needs of the geospatial community. During this time, code management within the project also evolved: The project used manual source code management from 1983 until 1999, when the Concurrent Versions System (CVS) [2] was introduced for revision control. Since 2007 the code management is based on Apache Subversion (SVN) [3] hosted by OSGeo [4], with migration to Git currently being worked on.

While version control, including the tracking of code submissions by individuals, evolved over time, the capabilities of the GRASS GIS software to provide user-sided automated citation have not kept up with the current advances in software citation. A standard GRASS GIS 7.4 installation is only capable to generate a BibTeX citation through the g.version module [5], which credits the GRASS Development Team as authors of the whole GRASS GIS software system.

The term GRASS Development Team summarizes the community of individuals which have developed and maintained the GRASS code base in past and at present. Within the team, individuals have been and are taking on varying roles when interacting with the code base, including, but not limited to original developer and maintainer. All roles are significant to the GRASS GIS project and should receive recognition (for both the team and the individual efforts) by due credit when scientific results based on these efforts are published. Since GRASS GIS has been under continuous development for over three decades, for many long established GRASS GIS modules the number of persons involved in code maintance, extension and refactoring already exceeds significantly the number of initial authors. A visual summary and overview of the development activities of the GRASS GIS codebase from 1999 to 2013 is showing different authors contributing in different periods of time and to different parts of the code[6].

To acknowledge the efforts that the members of the GRASS Development Team dedicated to specific modules or libraries, it is necessary to extend the GRASS GIS software by code-citation capabilities at the level of the individual functionalities, which are implemented as GRASS GIS modules.

Additionally, the development of best practices for software citation, especially metadata management, as currently being driven by communities like FORCE11 [7] or CodeMeta [8] remain to be acknowledged and adopted by the GRASS GIS community. This would allow to give credit to all stakeholders in the GRASS Development Team by state-of-the-art scientific citation practices.

The role of the OSGeo Foundation

The OSGeo Foundation [9] is an umbrella organisation which serves as a communication platform for a growing number of community driven geospatial open source projects since its founding in 2006. GRASS GIS, which preceded OSGeo by over two decades, was one of the founding projects and has ever since played an active role in shaping and advancing the OSGeo workflows and best-practices. The foundation has established common quality standards and best practices for projects, including social aspects of community governance and communication, but also technical aspects like coding standards and repository management. One central factor is the OSGeo incubation process, which is required for open source projects to become accredited within OSGeo. It is similar to the Apache Foundation graduation process [10], and assesses the maturity of project processes, and their compliance with the values and standards of OSGeo.

OSGeo embraces and fosters the paradigms of open source, open data, open standards and open education as the building blocks for open science [11]. The foundation belongs to the signatories of the commitment statement of the Enabling FAIR Data project [12] to enable FAIR data (including scientific software to work with the data) in earth, space and environmental science. OSGeo is commited to extend its support for the FAIR, i.e. findable, accessible, interoperable and reusable, principles [13]. However, software citation remains to be included in the OSGeo best practices.

Software development in GRASS GIS

While many functions provided through existing GRASS GIS modules have remained unchanged in the perception of the users, the portfolio of functionalities which are provided by the GRASS GIS software continues to grow. Contribution of new functionalities, frequently triggered by science projects, results in additions to the GRASS GIS codebase. This requires a sequence of actions, which are related to code quality and license, access and repository management aspects: The code which implements the algorithm for the new functionality migrates over time from the author's personal domain (i.e., his or her local computing environment), to the community domain of the GRASS project for code review and long term curation, paralleled by public access in the open access domain.

If the functionality provided by the code proves to be significant to the overall project, the code is migrated into the development branch of the GRASS codebase as a core module, to become a part of the next official GRASS release. This migration process is paralleled by iterative code quality assessment and improvement by the project community by public discussion, thorough review, refactoring and documentation according to the quality standards of the GRASS GIS project, in accordance to the best-practices of the greater OSGeo software ecotope.

Once a new GRASS module has reached add-on module status, the GRASS add-on discovery functionality provided by the module g.extension [14] to install add-on modules makes it both discoverable and accessible to the global user community, allowing for large-scale reuse, preventing waste of third party resources by redundant re-implementations and also potentially allowing to give credit to the developer(s) by citation.

When a functionality has become part of the main branch of the codebase, the task of code maintenance shifts from the original author to the GRASS Development Team. Participation of the authors(s) in the continuing maintenance and improvement effort is still appreciated, but no longer

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 1 April 2019 doi:10.20944/preprints201904.0008.v1

mandatory. The implemented functionality will continue to be maintained by the GRASS Development Team even after the original author(s) have left the project. Over time, such well maintained and iteratively updated code can reach levels of structuring and performance beyond the programming skills of the original authors.

This is similar to paradox of the ship of Theseus [15], which raises the question, if a wooden boat, which has had all physical parts replaced over time, is still identical to the vessel which was initially laid down. From the perspective of both the users and the GRASS Development Team, this is highly desirable and beneficial to the GRASS GIS project: In analogy to the ship of Theseus, the GRASS GIS project keeps rejuvenating its aging codebase in the face of evolving best practices in Information Technologies (IT) and also extends its tonnages by the growing number of included functionalities. The GRASS code repository ensures that all iterations of the GRASS GIS software (e.g. the many instances of Theseus ship) are kept available for future review and analysis.

Reward strategies in Science and Software Communities

Scientists, which base their research code on GRASS GIS must decide on a strategy if and how to publish their code. This currently results in conflicts regarding the quality of reward, code maintenance and reuse by others.

The first strategy, already described above, involves publishing the code as a new GRASS module in the GRASS GIS code repository. This strategy results in potentially widespread re-use, long-term maintenance and appreaciation by the GRASS community. However, the code author will only receive citation credit for his module if the user and prospective author of a scientific publication using the module undertakes the effort to manually derive a relevant citation from the credits on the module's manual page. In this case, it is also unlikely that members of the GRASS Development Team who contributed to the code updates ensuring its long term usability, will receive any due credit, as most traditional citation standards do not cover this type of critically important contributions.

The second strategy for the code author is to publish his or her novel GRASS-based code in an established scientific repository outside the GRASS GIS code repository, like those listed in the registry of research data repositories [16]. These repositories allow for reliable scientific citation through permanent persistent digital identifiers (PID), like Digital Object Identifiers (DOI) [17] to reference the landing pages, instead of transient URL links to module man pages, as currently used by GRASS GIS and other OSGeo projects.

However, from the established long-term expectation for fitness for use by the GRASS community, the second approach must be considered as "dead from the start": The task to further maintain the code in the chosen scientific repository must be shouldered entirely by the original authors, without the option of the GRASS Development Team to take over at some point. If the original developers will cease to support the maintenance of their submitted code within relatively short time the code archived in the repository will fossilize. Without regular updates the code will lose compatibility with future releases of the GRASS GIS code base and it will need major updates or re-implementations to make it executable in the future.

The GRASS GIS g.extension [14] module, which allows to integrate add-on modules to an existing GRASS GIS installation, provides the means to access GRASS add-on code from external code repositories, including RE3Data-listed scientific repositories [16] like Zenodo [18]. However, this requires existing prior knowledge by the prospective user where the particular module is stored and what it does. In addition, it is left to the user to assess the compatibility and trustworthiness of such unmaintained code in regard to the version of GRASS GIS currently being used. Since this applies to each user wishing to reuse the code, this can lead to repeated re-implementations over time.

g.citation: Software citation for GRASS GIS modules

The new g.citation module [19] complements the existing citation capabilities for GRASS GIS (g.version module) by supporting multiple scientific citation options at the granularity of particular

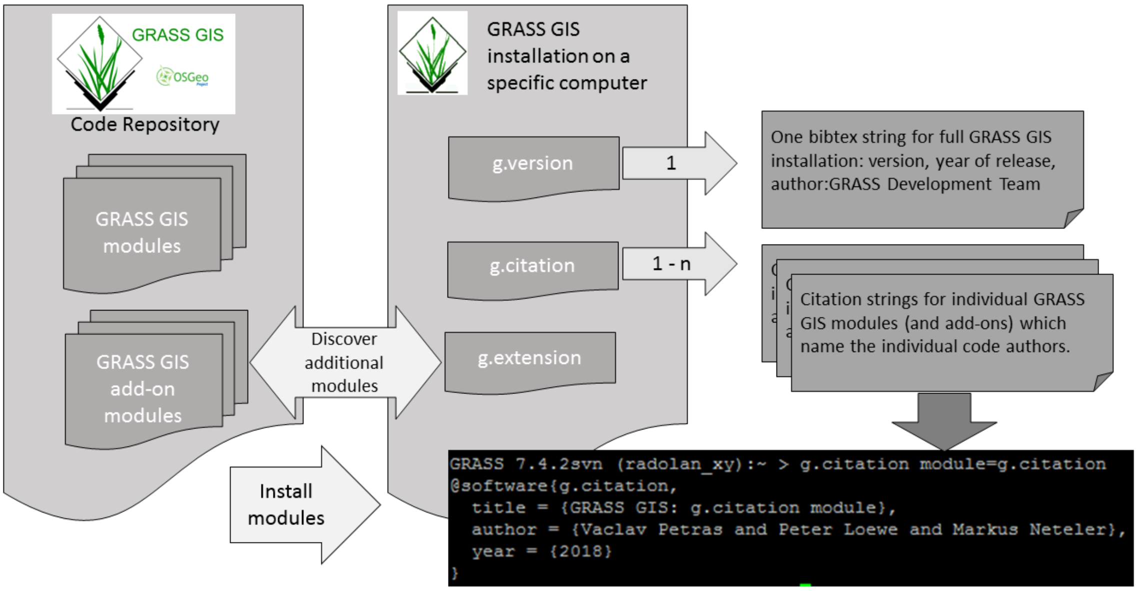

Figure 1. Overview of citation options for GRASS GIS: The GRASS GIS module g.version only provides a BibTeX citation string for the whole GRASS GIS installation, citing the GRASS Development Team as author. The new g.citation module can be installed from the GRASS GIS code repository via the g.extension module. It provides citations for individual GRASS GIS modules (and add-ons) in multiple output formats. The BibTeX output of self-referential application is shown in the lower right of the figure.

GRASS modules (Figure 1) in an automated and user-friendly way (Figure 2). This is a first step to overcome the current limitations of the GRASS GIS software regarding convenient and flexible citation capabilities to encourage users to cite both software and code, and to increase the motivation for code submissions to the GRASS GIS code repository for scientists. While the development of the module is currently in its late experimental phase in the GRASS sandbox code repository [20], it already supports three distinct categories of citation options.

The first category are citation strings formatted for human use in a text processor, according to the formatting rules (e.g. Figure 3). The second category provides machine readable generalized software metadata such as Citation File Format [21] (CFF) (Figure 4). The third category provides well formatted strings as input for reference management software used by humans or computer systems for formatting lists of references including a BibTex style (Figure 1) and Citation Style Language output[22] which are to be rendered by reference management tools and CSL-processors into a variety of citation styles, similar to citation rendering services already provided by scientific data repositories and citation infrastructures, as provided through the web portals of Zenodo or DataCite [23].

Next Steps

In addition to the improvement and extension of the g.citation module functionality and code quality, several tasks related to the GRASS GIS project and the OSGeo foundation have been identified, which can now be taken on because of the availability of g.citation.

The first task concerns the homogenization and improvement of the metadata within the GRASS GIS project: Currently, the quality of code-related metadata provided as human-readable content on the manual pages of GRASS GIS modules is mixed in terms of identifying contributions by individuals and their respective roles (e.g. original authors, maintainers, etc.). While best practices exist, a controlled vocabulary to describe the roles of members of the GRASS GIS Development Team has not been

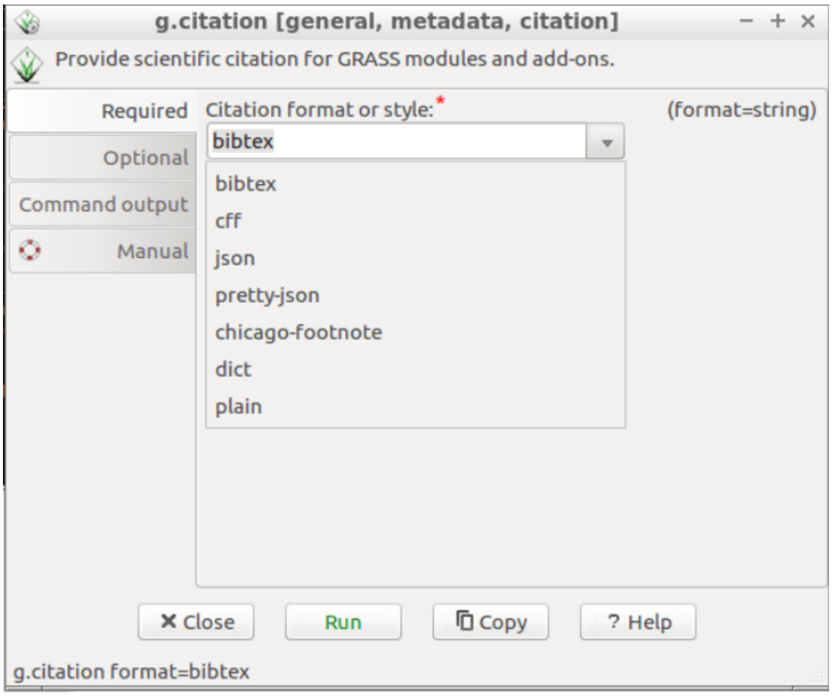

Figure 2. Screenshot of the GUI of the g.citation module, showing the currently included citation style options.

g.citation module=g.citation format=chicago-footnote

```
Vaclav Petras, Peter Loewe, and Markus Neteler,
GRASSS GIS module g.citation (7.4.2svn),
computer software (2018).
```

Figure 3. Example output for human readable output adhering to the Chicago footnote citation style, to be used with word processors

defined yet. Additionally, no machine-readable metadata are provided on the module manual pages. This makes it computationally hard to derive well-formed citation strings which now need to be parsed using different heuristics. To mitigate this, the output from g.citation derived from the current GRASS GIS manual pages can be used as input for a clean-up effort to homogenize and improve the structuring of the content of already existing GRASS GIS manual pages: As a follow-up step, it is intended to improve the GRASS GIS-internal code and documentation management workflow by integrating a new layer of structured CFF-files with well-defined metadata attributes as the source for HTML manual pages. This will result in an improved discoverability and scientific credit for content in the GRASS GIS code repository.

The second task is to establish code citation capabilities as a best practice for the OSGeo foundation. Once g.citation becomes included into the main branch of the GRASS GIS codebase, GRASS GIS can become a role model within OSGeo for code citation. In a follow up step, OSGeo can elect to include the topic of code citation capabilities and best practices into incubation process check-list.

The third task is to evolve the software repositories of OSGeo software projects to meet the current requirements of scientific data repositories and to establish them as recognized scientific infrastructure. This will involve updates on the metadata schemata, inclusion of machine-readable metadata to improve discoverability from outside and replacement of potentially transient (i.e., expected to fail on the very long-term time scale of literature and libraries) URLs with persistent identifiers. Many aspects of the GRASS GIS project infrastructure already comply with these requirements, like the structuring of manual page content for GRASS modules, whose human readable content already meets

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 1 April 2019 doi:10.20944/preprints201904.0008.v1

g.citation module=g.citation format=cff

```
cff-version: 1.0.3
message: "If you use this software, please cite it as below."
authors:
  - family-names: Petras
    given-names: Vaclav
    orcid: 0000-0001-5566-9236
  - family-names: Loewe
    given-names: Peter
  - family-names: Neteler
    given-names: Markus
title: "GRASS GIS: g.citation module"
version: 7.4.2svn
date-released: 2018-08-06
license: GPL-2.0-or-later
```

Figure 4. Example output, both human and machine readable, in code citation format to be used for reference management software (used by humans) or as input for machine actionable citation harvesting by entities like DataCite.

the requirements for landing pages for DOI-referenced data sets. The GRASS GIS project could also become the driver for this within OSGeo.

Conclusion

The GRASS GIS add-on module g.citation extends the existing functionality of GRASS GIS by generating human- and machine-readable citation information for individual GRASS GIS modules. This allows scientists to give the due credit to the respective authors of these individual modules. The new functionality is a pragmatic step towards improvements of workflows and infrastructures within the GRASS GIS project, which can become examples for the greater OSGeo community. Based on this relatively small step, follow up efforts, which can have positive effects on larger scales, can be undertaken to homogenize the quality of metadata within the existing GRASS GIS codebase, establish g.citation as a OSGeo-wide reference implementation, and make code citation capabilities a topic for the OSGeo incubation process.

Author Contributions: Conceptualization and Methodology, Peter Loewe, Vaclav Petras, Markus Neteler and Helena Mitasova; Software, Vaclav Petras and Markus Neteler; Data Curation and Funding Acquisition, Peter Loewe; Supervision and Resources, Helena Mitasova

Funding: "This research received no external funding." "The Article Processing Charge (APC) was funded by Society of Friends of DIW Berlin (VdF)".

Conflicts of Interest: "The authors declare no conflict of interest." "The funders had no role in the writing of the manuscript, or in the decision to publish the results".

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 1 April 2019 doi:10.20944/preprints201904.0008.v1

Abbreviations

The following abbreviations are used in this manuscript:

CFF	Citation File Format
CSL	Citation Style Language
CVS	Concurrent Versions System
DOI	Digital Object Identifier
FAIR	Findable, Accessible, Interoperable, Reusable
FORCE11	Future of Research Communications
GIS	Geographic Information System
GRASS GIS	Geographic Resources Analysis Support System
IT	Information Technology
JSON	Javascript Object Notation
OSGeo	Open Source Geospatial Foundation
PID	Persitent Identifier
RE3Data	Registry of Research Data Repositories
SVN	Apache Subversion
URL	Uniform Resource Locator

References

1. GRASS GIS community. GRASS GIS - Bringing advanced geospatial technologies to the world. https://grass.osgeo.org, 2018. [Online; accessed 13-September-2018].
2. CVS Team. Concurrent Versions System. https://en.wikipedia.org/wiki/Concurrent$_{V}$*ersions*$_{S}$*ystem*, 2018.[*Online*; *accessed*13 – *September* – 2018].
3. Apache Foundation. Apache Subversion. https://subversion.apache.org, 2018. [Online; accessed 13-September-2018].
4. GRASS GIS community. GRASS GIS Software Repository on SVN. https://svn.osgeo.org/grass/, 2018. [Online; accessed 13-September-2018].
5. GRASS GIS Development Team. GRASS GIS: g.version module. https://grass.osgeo.org/grass74/manuals/g.version.html, 2018. [Online; accessed 13-September-2018].
6. Neteler, M. GRASS GIS 6.4 development visualization from 1999 to 2013. https://doi.org/10.5446/14652, 2013. [Online; accessed 13-September-2018].
7. FORCE11 Community. FORCE11 – Future Research Communication and e-Scholarship, 2011. [Online; accessed 13-September-2018].
8. CodeMeta Project. CodeMeta Website. https://codemeta.github.io/, 2018. [Online; accessed 13-September-2018].
9. OSGeo Foundation. OSGeo Website. https://www.osgeo.org/, 2006. [Online; accessed 13-September-2018].
10. Apache Foundation. Apache Website, 2018. [Online; accessed 13-September-2018].
11. Wikipedia contributors. Open Science — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/wiki/Open$_{s}$*cience*, 2012. [*Online*; *accessed*13 – *September* – 2018].
12. Enabling FAIR Data Project. Enabling FAIR Data Project Website. http://www.copdess.org/enabling-fair-data-project, 2016. [Online; accessed 13-September-2018].
13. Enabling FAIR Data Project. Enabling FAIR Data Commitment Statement in the Earth, Space, and Environmental Sciences. http://www.copdess.org/enabling-fair-data-project/commitment-to-enabling-fair-data-in-the-earth-space-and-environmen 2018. [Online; accessed 13-September-2018].
14. Markus Neteler, Martin Landa and Vaclav Petras. GRASS GIS: g.extension module. https://grass.osgeo.org/grass76/manuals/g.extension.html, 2017.
15. Wikipedia contributors. Ship of Theseus – Wikipedia, The Free Encyclopedia, 2012. [Online; accessed 13-September-2018].
16. re3data Project. re3data website. https://www.re3data.org/, 2018. [Online; accessed 13-September-2018].
17. DOI community. DOI website. https://www.doi.org/, 2018. [Online; accessed 13-September-2018].
18. Zenodo Community. Zenodo research data repository. https://www.zenodo.org/, 2015. [Online; accessed 13-September-2018].

 doi:10.20944/preprints201904.0008.v1

19. Petras, V.; Löwe, P.; Neteler, M. GRASS GIS: g.citation module. https://svn.osgeo.org/grass/sandbox/wenzeslaus/g.citation/g.citation.html, 2018.
20. Petras, V.; Löwe, P.; Neteler, M.; Mitasova, H. GRASS GIS: g.citation preliminary code repository. https://trac.osgeo.org/grass/browser/sandbox/wenzeslaus/g.citation, 2018. [Online; accessed 13-September-2018].
21. Druskat, S. Citation File Format (CFF) GitHub repository. https://citation-file-format.github.io/, 2018. [Online; accessed 13-September-2018].
22. CSL team. Citation Style Language Website. https://citationstyles.org/, 2018. [Online; accessed 13-September-2018].
23. Datacite team. Datacite Website. https://www.datacite.org/, 2018. [Online; accessed 13-September-2018].