Software Citation: Principles, Discussion, and Metadata

Daniel S. Katz
Associate Director for Scientific Software & Applications, NCSA
Research Associate Professor, ECE
Research Associate Professor, iSchool
dskatz@illinois.edu, d.katz@ieee.org, @danielskatz

Principles work: with Arfon M. Smith, Kyle E. Niemeyer & WG
Software citation motivation

- Research process is increasingly digital
  - Research outputs/products not just papers and books
  - Also **software**, data, slides, posters, graphs, maps, etc.
  - Research knowledge is embedded in these components
  - Papers too are more digital; can be executable and reproducible
- But citation system was created for papers/books
- We need to either/both
  1. Jam software into current citation system
  2. Rework citation system
    - Focus on 1 as possible; 2 is very hard.
- Challenge: not just how to identify software in a paper
  - How to identify software used within research process
Software citation principles: People

- **FORCE11 Software Citation working group (Arfon Smith & Dan Katz)**
  - Started July 2015, ~40 members
  - Working on GitHub: https://github.com/force11/force11-scwg
  - Also listed on FORCE11: https://www.force11.org/group/software-citation-working-group

- **WSSSPE3 Credit & Citation working group (Kyle Niemeyer)**
  - September 2015

- **WSSSPE3 group merged into FORCE11 group in Oct.**
  - ~55 members (researchers, developers, publishers, repositories, librarians)
Software citation principles: Process

- Review of existing community practices
  - Software Sustainability Institute, WSSSPE, Project CRediT, Ontosoft, CodeMeta
- Developed use cases (collaborative via Google Doc)
- Drafted software citation principles document
  - Started with data citation principles
  - Updated based on software use cases and related work
  - Updated based working group discussions, community feedback and review of draft, workshop at FORCE2016 in April
  - Document contents: 6 principles, motivation, summary of use cases, related work, and discussion (including recommendations)
- Working document in GitHub, linked from FORCE11 SCWG page, discussion has been via GitHub issues, changes have been tracked
Principles (1/3)

1. Importance: Software should be considered a legitimate and citable product of research. Software citations should be accorded the same importance in the scholarly record as citations of other research products, such as publications and data...

2. Credit and Attribution: Software citations should facilitate giving scholarly credit and normative and legal attribution to all contributors to the software, recognizing that a single style or mechanism of attribution may not be applicable to all software.
3. Unique Identification: A software citation should include a method for identification that is machine actionable, globally unique, interoperable, and recognized by at least a community of the corresponding domain experts, and preferably by general public researchers.

4. Persistence: Unique identifiers and metadata describing the software and its disposition should persist – even beyond the lifespan of the software they describe.
5. **Accessibility**: Software citations should permit and facilitate access to the software itself and to its associated metadata, documentation, data, and other materials necessary for both humans and machines to make informed use of the referenced software.

6. **Specificity**: Software citations should facilitate identification of, and access to, the specific version of software that was used. Software identification should be as specific as necessary, such as using version numbers, revision numbers, or variants such as platforms.
Discussion: What to cite

• Importance principle: “…authors should cite the appropriate set of software products just as they cite the appropriate set of papers”
• What software to cite decided by author(s) of product, in context of community norms and practices
• POWL: “Do not cite standard office software (e.g. Word, Excel) or programming languages. Provide references only for specialized software.”
• i.e., if using different software could produce different data or results, then the software used should be cited
Discussion: What to cite (citation vs provenance & reproducibility)

- Provenance/reproducibility requirements > citation requirements
- Citation: software important to research outcome
- Provenance: all steps (including software) in research
- For data research product, provenance data includes all cited software, not vice versa
- Software citation principles cover minimal needs for software citation for software identification
- Provenance & reproducibility may need more metadata
Discussion: Software papers

• Goal: Software should be cited
• Practice: Papers about software ("software papers") are published and cited
• Importance principle (1) and other discussion: The software itself should be cited on the same basis as any other research product; authors should cite the appropriate set of software products
• Ok to cite software paper too, if it contains results (performance, validation, etc.) that are important to the work
• If the software authors ask users to cite software paper, can do so, in addition to citing to the software
Discussion: Derived software

• Imagine Code A is derived from Code B, and a paper uses and cites Code A
• Should the paper also cite Code B?
• No, any research builds on other research
• Each research product just cites those products that it directly builds on
• Together, this give credit and knowledge chains
• Science historians study these chains
• More automated analyses may also develop, such as transitive credit

Discussion: Software peer review

• Important issue for software in science
• Probably out-of-scope in citation discussion
• Goal of software citation is to identify software that has been used in a scholarly product
  • Whether or not that software has been peer-reviewed is irrelevant
• Possible exception: if peer-review status of software is part of software metadata
• Working group: this is not needed to identify the software
Discussion: Citations in text

• Each publisher/publication has a style it prefers
  • e.g., AMS, APA, Chicago, MLA
• Examples for software using these styles published by Lipson
• Citations typically sent to publishers as text formatted in that citation style, not as structured metadata
• Recommendation
• All text citation styles should support:
  • a) a label indicating that this is software, e.g. [Computer program]
  • b) support for version information, e.g. Version 1.8.7
Discussion: Citation limits

• Software citation principles
• → more software citations in scholarly products
• → more overall citations
• Some journals have strict limits on
  • Number of citations
  • Number of pages (including references)
• Recommendations to publishers:
  • Add specific instructions regarding software citations to author guidelines to not disincentivize software citation
  • Don’t include references in content counted against page limits
Discussion: Unique identification

- Recommend DOIs for identification of published software
- However, identifier can point to
  1. a specific version of a piece of software
  2. the piece of software (all versions of the software)
  3. the latest version of a piece of software
- One piece of software may have identifiers of all 3 types
- And maybe 1+ software papers, each with identifiers
- Use cases:
  - Cite a specific version
  - Cite the software in general
  - Cite the latest version
  - Link multiple releases together, to understanding all citations
Discussion: Unique identification (cont.)

• Principles intended to apply at all levels
• To all identifiers types, e.g., DOIs, RRIDs, ARKS, etc.
• Though again: recommend when possible use DOIs that identify specific versions of source code
• RRIDs developed by the FORCE11 Resource Identification Initiative
  • Discussed for use to identify software packages (not specific versions)
  • FORCE11 Resource Identification Technical Specifications Working Group says “Information resources like software are better suited to the Software Citation WG”
• Currently no consensus on RRIDs for software
Discussion: Types of software

- Principles and discussion generally focus on software as source code
- But some software is only available as an executable, a container, or a service
- Principles intended to apply to all these forms of software
- Implementation of principles will differ by software type
- When software exists as both source code and another type, cite the source code
Discussion: Access to software

- Accessibility principle: “software citations should permit and facilitate access to the software itself”
- Metadata should provide access information
- Free software: metadata includes UID that resolves to URL to specific version of software
- Commercial software: metadata provides information on how to access the specific software
  - E.g., company’s product number, URL to buy the software
- If software isn’t available now, it still should be cited along with information about how it was accessed
Software citation principles: Status & next steps

- Draft document published on FORCE11 website
- Two week open comment period (ended May 20)
- Updated version will be published on FORCE11 website and on archive site (e.g., F1000Research, figshare, Zenodo)
- Endorsement period for both individuals and organizations
- Will create infographic and 1–3 slides
- Software Citation Working Group ends
- Software Citation Implementation group starts? (work with institutions, publishers, funders, researchers, etc. - and write implementation examples paper?)