

Software Citation Use Cases

The goal of this document is to build a list of use cases and a short description of each. This will be used both to properly scope the activities of the Software Citation Working Group moving forward, and to support the Software Citation Principles document.

Please either add any missing use cases, or elaborate existing descriptions.

Codemeta Working group, April 16th, 2016 PM, discussion document available here

https://docs.google.com/document/d/1SalOP_tLw7FC6jnoz5hJBB2jA7xBCIKqc88wgmJu5Ro/edit#

TOC:

Software Citation Use Cases

Summary table

Stakeholders

Definitions

Use cases

1. Researcher who uses someone else's software for a paper
2. Researcher who uses someone else's software for new software
3. Researcher who contributes to someone else's software (open source project)
4. Researcher who wants to know who uses the researcher's software
5. Researcher who wants to publish about a piece of software
6. Publisher who wants to publish papers that cite software
7. Indexer (e.g., Scopus, WoS, Scholar, MS Academic Search) who wants to build a catalog of software
8. Domain group (e.g., ASCL, bioCADDIE), Libraries, and Archives (e.g., University library, laboratory archive, etc.) who wants to build a catalog/registry of institutional or domain software
9. Repository showing scientific impact of holdings
10. Funder who wants to know how software they funded has been used
11. Researcher gets credit for software development at the academic/governmental institution, in his/her academic/professional career, etc.
12. Researcher who wants to "reproduce" another person/group's analysis
13. Researcher who benchmarks someone else's software with or without modification on one or many hardware platforms for publication
14. Reference management system used by researchers to author a manuscript
15. Researcher who wants to find a piece of software to implement a task
16. Researcher wants to record the software that generated some data

Summary table

Stakeholder	Use/ wants to	Basic Metadata								
		UID	SW name	Author (s)	Contributor role	Version #	Rel. date	Location/ repo	Indexed citations	SW license
Researcher	1. Use someone else's SW for a paper	●	●	●		●	●	●		
Researcher	2. Use someone else's SW for new SW	●	●	●		●	●	●		○
Researcher	3. Contribute to SW	●	●	●	●	●	●	●		○
Researcher	4. Find citations of SW	●	●						●	
Researcher	5. Get credit for SW development	●	●	●	●		●	●		○
Researcher	6. "Reproduce" analysis	●	●			●	●	●		○
Researcher	7. Benchmark SW	●	●	●		●	●	●		○
Researcher	8. Find SW to implement task	●	●	●				●	●	○
Publisher	9. Publish SW paper	●	●	●		●	●	●		
Publisher	10. Publish papers that cite SW	●	●	●		●	●	●	●	
Indexer	11. Build index of software and citations of it	●	●	●		●	●	●	●	○
Domain group/ Library/ archive	12. Build SW catalog/registry	●	●	●				●		
Repository	13. Show scientific Impact of holdings	●	●						●	
Funder	14. Show how funded SW has been used	●	●						●	
Evaluator	15. Evaluate researcher contributions	●		●	●		●		●	
Citation manager	16. Store software entry	●	●	●		●	●	●	●	
Repository	17. Publish mixed	●	●	●		●	●	●		○

	data/software package								
--	-----------------------	--	--	--	--	--	--	--	--

Stakeholders

Researcher: includes both academic researchers (e.g., postdoc, tenure-track faculty member) and research software engineers.

Publisher: includes both traditional publishers that publish text and/or software papers as well as archives such as Zenodo that directly publish software.

Repository: refers to public software repositories such as [Astronomy Source Code Library \(ASCL\)](#), GitHub, Netlib, Comprehensive R Archive Network (CRAN), institutional repositories.

Funder: is a group that funds software or work using software.

Indexer: examples include Scopus, Web of Science, Google Scholar, and Microsoft Academic Search.

Domain group: includes [bioCADDIE](#), [Computational Infrastructure for Geodynamics \(CIG\)](#), etc.

Citation Manager Developer: people that create software that researchers use to manage citation information, and semi-automatically insert those citations into research products

Definitions

Unique identifier: refers to identifiers such as a DOI, ARK, or PURL, must also be persistent and machine-actionable.

Reproduce: can mean reproduction, replication, verification, validation, repeatability, and/or utility.

Use cases

1. Researcher who uses someone else's software for a paper

One of the most common use cases may be researchers who use someone else's software and want to cite it in a technical paper. This will be similar to existing practices for citing research artifacts in papers.

“Requirements” for researcher:

- Name of software
- Names of software authors/contributors
- Software version number and release date, or download date
- Location/repository, or contact name/email (if not publicly available)
- Citable DOI of software
- Format for citing software in text and in bibliography

Possible steps:

0. Software developers create CITATION file and associate with source code release/repository.
1. Researcher finds and uses software for research paper.
2. Researcher identifies citation metadata file (e.g., “CITATION” file) associated with downloaded/installed software source code or in online repository/published location.
 - a. CITATION file includes necessary citation metadata
 - b. CITATION file may include BibTeX entry, suggested citation format
3. Researcher cites software appropriately, e.g. in methodology section; reference included in bibliography.

Need to add more cases for this use case, e.g. what happens if the code is commercial, if it comes from a package manager (e.g., binary, macports), etc?

<https://github.com/force11/SCWG-impl-paper>

2. Researcher who uses someone else’s software for new software

In this case, a researcher develops new software that incorporates or depends on existing software. In order to credit the developer(s), the researcher will include citations in his/her source code, documentation, or other metadata in a similar manner to papers

Requirements for researcher:

- Name of software
- Names of software authors/contributors
- Software version number and release date
- Location/repository
- Citable DOI of software
- Format for citing software in source code, documentation, or citation metadata file

Possible steps:

0. Assume that software developers have created a CITATION file and associated with the source code release/repository.
1. Researcher finds and uses software in the development of new software.
2. Researcher identifies citation metadata file (e.g., “CITATION” file) associated with downloaded/installed software source code or in online repository/published location.
 - a. CITATION file includes necessary citation metadata
 - b. CITATION file may include BibTeX entry, suggested citation format

3. Researcher cites software in source code, documentation, or other metadata-containing file.

3. Researcher who contributes to someone else's software (open source project)

A researcher wants to contribute to someone else's software in the manner in which their contributions will be accepted and recognized.

Possible steps:

1. Researcher finds information about the software, and how contributors will be recognized
2. Researcher possibly submit a Contributor License Agreement (CLA) or Copyright Assignment Agreement (CAA) to allow the contributed content to be distributed with the software being contributed to
3. Researcher contributes to the software
4. Software maintainers accept contribution, recognize researcher's contribution, and update the software metadata as appropriate

4. Researcher who wants to know who uses the researcher's software

This case is similar to a researcher who wants to find other papers/publications that cite a particular paper.

A researcher wants to gauge the usage of her software within or across communities and measure its impact on research for both credit and funding.

Requirements:

- Uniquely identify software
- Indexed citations of software
- Indexed papers that use software

Steps:

1. Researcher finds software official name or unique DOI in metadata associated with downloaded/installed source code or in online repository/published location.
2. Researcher searches for software, may use online indexer (e.g., Scopus, Web of Science, Google Scholar) using software name or DOI.
3. Online indexer presents entry for software with list of citations, if any.
 - a. Ideally, entry will also include metadata contained in software CITATION file and citation example.

5. Researcher gets credit for software development at the academic/governmental institution, in his/her academic/professional career, etc.

Requirements for researcher:

- Name of software
- Names of software authors/contributors
- Location/repository
- Citable DOI of software
- Format for citing software in an official CV, in a departmental/institutional review report, etc.
- Role in the software creation, that is linked to version or component

Role in contributing to the software as a ‘package’ (not just lines of code) development of benchmarks, testing, documentation, tutorials etc.

-

6. Researcher who wants to “reproduce” another person/group’s analysis

When a researcher wants to understand or verify a research results from another researcher, they would like to use the same software. Note that accessing the exact same software is necessary but not sufficient for reproducibility.

Requirements for researcher:

- Name of software
- Location/repository for the exact release that was used
- DOI or other persistent handle for that specific release

Release has all components necessary for reproducing the work (note that this ideally also means sample inputs and outputs)

-

7. Researcher who wants to find a piece of software to implement a task

This is the case where a research is looking for software to use but wants to understand whether it is being used in a scholarly fashion. For example, a researcher searches through a software repository and finds a package that might be useful. They look to find whether it has been used by others in the scientific literature.

Requirements

- Either the software documentation page has a reference to existing literature that makes use of it.
- There is a mechanism to look it up

8. Publisher who wants to publish software paper

This case asks what information regarding software is needed for a publisher who wants to publish a paper describing that software.

Requirements for publisher:

- Name of software

- Names of software authors/contributors
- Location/repository
- Citable DOI of software
- Format for citing software in JATS, for example, as well as references in the text itself

9. Publisher who wants to publish papers that cite software

This case asks what information regarding software is needed for a publisher who wants to publish papers that cite that software.

Requirements for publisher:

- Name of software
- Names of software authors/contributors
- Location/repository
- Citable DOI of software
- Format for citing software in JATS, for example, as well as references in the text itself

10. Indexer (e.g., Scopus, WoS, Scholar, MS Academic Search) who wants to build a catalog of software

Provide an index over the software that is used within the research domain. Track how that software is being used by different groups of researchers and to what ends.

Requirements:

- Uniquely identify pieces of software used by the research literature
- Connect authors and organizations to that software
- Connect various software versions together

11. Domain group (e.g., ASCL, bioCADDIE), Libraries, and Archives (e.g., University library, laboratory archive, etc.) who wants to build a catalog/registry of institutional or domain software

- There are two different examples here: One is building a catalog/archive of software produced by those affiliated with the institution
- The other is along the lines of Sayeed Choudhury's note that "data are the new special collections". An institution may choose to build a catalog/archive of many things within a single topic or subject in order to secure all the software on a certain topic or build a collection that may draw users to their establishment, much like special collections now do for university libraries and archives.

12. Repository showing scientific impact of holdings

A repository that archives and/or maintains a collection of software. The repository would like to address usage and impact of software in its holding. Usage would aid potential users whether the software is being actively maintained or developed or has been superseded. Both would help repository know how to direct resources e.g. maintenance, training etc.

Requirements:

- Code name ideally, a unique identifier
- Relationships to previous versions
- Connect to repository
- Connect to research

similar to Funder case

- mineable via repository name, code name, “doi”, or citation.

13. Funder who wants to know how software they funded has been used

- steps/requirements
-

14. Evaluator or funder wants to evaluate contributions of a researcher.

1.

15. Reference management system used by researchers to author a manuscript

Reference management systems may need to be updated to internally understand that their is a software reference type, and to be able to output references to software in common formats.

Requirements for reference manager:

- Names of software authors/contributors
- Software version number and release date
- Location/repository
- Citable DOI of software or paper recommended for citation
- Format for citing software in citation metadata file
- Citation metadata tags embedded in DOI landing page/software project page for easy ingest

Possible steps:

1. Reference management system such as EndNote, Mendeley, Zotero, etc. builds affordances for software references.
2. Researcher finds software citation and adds it to their reference manager library.
 - a. by importing from the CITATION file (e.g., BibTeX, RIS)
 - b. by clicking on, e.g., an “add to Zotero library” widget in web browser
3. Researcher writes a paper and uses the reference manager to generate citations or bibliography.
-

Not adopted in use cases table:

Researcher who benchmarks someone else's software with or without modification on one or many hardware platforms for publication

This case describes the need for a research who has contributed to software (by design, software engineering, development, testing, patching, documentation, training, evangelizing, etc.) to have their software work recognized by their employer or colleagues for the purpose of career advancement and increased professional reputation.

Requirements for researcher:

- Name of software
- Names of software authors/contributors
- Software version number and release date
- Location/repository
- Citable DOI of software or paper recommended for citation
- Format for citing software in source code or citation metadata file

Possible steps:

1. Software developers create CITATION file and associate with source code release/repository.
2. Researcher finds and uses software in the development of new software.
3. Researcher identifies citation metadata file (e.g., "CITATION" file) associated with downloaded/installed software source code or in online repository/published location.
 - a. CITATION file includes necessary citation metadata
 - b. CITATION file may include BibTeX entry, suggested citation format
4. Researcher cites software in source code, documentation, or other metadata-containing file.

Researcher who wants to publish about a piece of software

The research wants to publish about a version of software they have produced. A key part of this use case is to be able to connect the given narrative to a specific version of the software in questions and connect that in large story.

Requirements:

- Name of software
- Names of software authors/contributors
- Location/repository
- Citable DOI of Software

Links to older versions of software

Researcher wants to record the software that generated some data

This is the case where a researcher is using some software to perform an analysis, either of a physical sample or of data. The researcher needs to know which version was used, for example in case a bug was fixed. Note that knowing the software and its version is not sufficient to determine the “conditions” of the analysis, but they are essential.

Requirements

- The analysis, or the generated data, has information about the software used.

Researcher who wants to reproduce experience of use of a particular software implementation in context

Researcher is engaged in historical/cultural research -- e.g. study of video games as cultural artifacts.

Requirements

- Name of software
- Software version number
- Documentation of the execution environment/context
- Location/repository for virtual machine (or equivalent) comprising both software and execution environment/context
- Persistent identifier associated with virtual machine instance (or equivalent) comprising both software and execution environment/context

Example of steps

- Researcher obtains persistent ID from citation
- Research uses a persistent ID resolution service to resolve ID to a location of an executable VM instance in a repository
- Researcher obtains VM in the repository, executes it, and interacts with software

References:

- Use case examples and commentary
 - Library of congress report on software preservation:
<http://www.digitalpreservation.gov/meetings/preservingsoftware2013.html>
 - CASPAR Report on Software Significant Properties
<http://sigsoft.dcc.rl.ac.uk/twiki/pub/Main/SigSoftTalks/SignificantPropertiesofSoftware-r1.doc>
- Examples of VM archiving and repositories
 - <https://olivearchive.org/>
 - <https://archive.org/details/internetarcade>
-

Contributor:

Micah Altman [@maltman] <escience@mit.edu>