How to make software fit in research citation graphs

Stephan Druskat
German Aerospace Center (DLR), Intelligent and Distributed Systems & Humboldt-Universität zu Berlin, Dept. of Computer Science

RSEConUK 2019, Birmingham
18 September 2019

stephan.druskat@dlr.de
@stdruskat
Aims

• **My aim:** Help integrate software in a more complete and fairer system of citation.

• **How:** Apply software engineering methods to create citation graphs for software and their dependencies.

• **This talk:** Show how modeling the output of citation can help understand the requirements for my work, and for the implementation of software citation.
Software, citation, and everything

• Software is a research product! [1, 2]
• Software citation principles! [3]
The citation system and its functions

A sociotechnical system which provides

- Context
- Trust & authority
- Recognition of value, credit (for individuals and groups/entities)
- Compliance
- Discursivity
- Reproducibility
The citation system and its functions

A sociotechnical system which provides

- **Context**: Understand how knowledge was established and is used
- **Trust & authority**
- **Recognition of value, credit (for individuals and groups/entities)**
- **Compliance**
- **Discursivity**
- **Reproducibility**
The citation system and its functions

A sociotechnical system which provides

- Context: Understand how knowledge was established and is used
- Trust & authority: Trust in research and researchers, authority over research
- Recognition of value, credit (for individuals and groups/entities)
- Compliance
- Discursivity
- Reproducibility

Photo by Liane Metzler on Unsplash
The citation system and its functions

A sociotechnical system which provides

- Context: Understand how knowledge was established and is used
- Trust & authority: Trust in research and researchers, authority over research
- Recognition of value, credit (for individuals and groups/entities)
- Compliance
- Discursivity
- Reproducibility

Photo by Riccardo Annandale on Unsplash
The citation system and its functions

A sociotechnical system which provides

• Context: Understand how knowledge was established and is used
• Trust & authority: Trust in research and researchers, authority over research
• Recognition of value, credit (for individuals and groups/entities)
• Compliance: with established rules of scholarly practice
• Discursivity
• Reproducibility
The citation system and its functions

A sociotechnical system which provides

- **Context**: Understand how knowledge was established and is used
- **Trust & authority**: Trust in research and researchers, authority over research
- **Recognition of value, credit**: (for individuals and groups/entities)
- **Compliance**: with established rules of scholarly practice
- **Discursivity**: through enabling epistemic change ("re-writing of the past")
- **Reproducibility**
The citation system and its functions

A sociotechnical system which provides

- Context: Understand how knowledge was established and is used
- Trust & authority: Trust in research and researchers, authority over research
- Recognition of value, credit (for individuals and groups/entities)
- Compliance: with established rules of scholarly practice
- Discursivity: through enabling epistemic change ("re-writing of the past")
- Reproducibility: by providing provenance of research, i.e., “what was used“
Software as a research product must be integrated in the citation system so that it can participate in all functions.
Modeling the citation system

Stage 1: Modeling the context function

- **Research citation graph:** A directed graph $G = (V, E)$
- $V$ are vertices representing research products
- $E$ are directed edges representing citation
Modeling the citation system

Stage 2: Modeling the social functions: trust & authority, credit (and evaluation)

• Add:
  • Authors (and authorship relations)
  • Affiliations (and affiliation relations)
  • “Product containers“: journals, repositories, archives, etc.
  (and published-in relations)
  \[ G = (V, E) \]
  \[ V = \{P, A, I, C\} \]
  \[ L : V \rightarrow V \] to set
  \[ L(v) = P \text{ when } v \in P \in V, \text{ etc.} \]

“Pre-digitalization research citation graph”
Modeling the citation system: factoring in software specifics

Software differs from textual research products in

• the form its artifacts can take,
• its notion of finality and the relationships between its artifacts,
• the citability of its concepts,
• its dynamicity,
• the containment relationships between a product and its contributions,
• the roles which contribute to it.
Modeling the citation system: factoring in software specifics

Software differs from textual research products in

• the form its artifacts can take: source code vs. binary artifacts
• its notion of finality and the relationships between its artifacts,
• the citability of its concepts,
• its dynamicity,
• the containment relationships between a product and its contributions,
• the roles which contribute to it.
Modeling the citation system: factoring in software specifics

Software differs from textual research products in

• the form its artifacts can take: source code vs. binary artifacts
• its notion of finality and the relationships between its artifacts: versions & seriality (vs. finality)
• the citability of its concepts,
• its dynamicity,
• the containment relationships between a product and its contributions,
• the roles which contribute to it.
Modeling the citation system: factoring in software specifics

Software differs from textual research products in

- the form its artifacts can take: source code vs. binary artifacts
- its notion of finality and the relationships between its artifacts: versions & seriality (vs. finality)
- the citability of its concepts: software concepts are citable and cited
- its dynamicity,
- the containment relationships between a product and its contributions,
- the roles which contribute to it.
Modeling the citation system: factoring in software specifics

Software differs from textual research products in

• the form its artifacts can take: source code vs. binary artifacts
• its notion of finality and the relationships between its artifacts: versions & seriality (vs. finality)
• the citability of its concepts: software concepts are citable and cited
• its dynamicity: passive & functionally active (states, execution paths)
• the containment relationships between a product and its contributions,
• the roles which contribute to it.
Modeling the citation system: factoring in software specifics

Software differs from textual research products in

• the form its artifacts can take: source code vs. binary artifacts
• its notion of finality and the relationships between its artifacts: versions & seriality (vs. finality)
• the citability of its concepts: software concepts are citable and cited
• its dynamicity: passive & functionally active (states, execution paths)
• the containment relationships between a product and its contributions: dependencies are part of the product, at runtime at the latest
• the roles which contribute to it.
Modeling the citation system: factoring in software specifics

Software differs from textual research products in

• the form its artifacts can take: source code vs. binary artifacts
• its notion of finality and the relationships between its artifacts: versions & seriality (vs. finality)
• the citability of its concepts: software concepts are citable and cited
• its dynamicity: passive & functionally active (states, execution paths)
• the containment relationships between a product and its contributions: dependencies are part of the product, at runtime at the latest
• the roles which contribute to it: testers, designers, bug reporters, etc.
Modeling the citation system: requirements

• **Compliance:** Updated funders‘ guidelines/good scholarly practice guidelines require software citation

• **Reproducibility:** Complete and correct citation of used product, software should also cite products it builds on, including other software

• Model must include
  • Versions (and precedence relations)
  • Concepts (and realization relations)
  • Different contribution types
A model of research citation graphs that include software

Stage 3: Model the missing functions

Compliance: software cites its references

Reproducibility: exact references are cited completely and correctly (allowing unique identification), from software and other products

Discursivity: potentially applicable to software citing its (non-software) references

\[
G = (V, E) \\
V = \{P, A, I, C, O\} \\
E = \{E_{\text{affil}}, E_{\text{cite}}, E_{\text{contrib}}, E_{\text{prec}}, E_{\text{pub-in}}, E_{\text{real}}\}
\]
Research citation graphs: applications

• The obvious stuff:
  back-tracking context exploration, citation tracking, tracking of concept citation, self-citation analysis

• The less obvious stuff:
  Contribution role analysis, analysis of software development practices

• The cool/important/overdue stuff:
  Credit for „hidden“ contributions to research, retrieval of transitive credit
Research citation graphs: transitive credit

- **Transitive credit [5]:**
  - Fractional credit for a research product is not distributed over authors alone, but also over referenced research products (credit map)
  - Credit maps for a product feed into the credit map for products that reference it
  - \( p_1 \) cites \( p_2 \), \( p_2 \) is awarded 20% credit for \( p_1 \rightarrow a_3 \) is awarded 50% credit for \( p_2 \rightarrow a_3 \) is awarded 10% credit for \( p_1 \)

- **Calculating fractional credit:**
  - For contributing humans: manually, augmented
  - For contributing dependencies: programmatically
    - Software engineering:
      - Call frequencies + complexity metrics
    - Enables evaluation methods for software dependencies
Instantiating research software citation graphs: challenges

• Cultural challenges:
  • Software as a research product (Importance principle, [3])
  • Practice of software citation
  • Unique identification of individuals and groups/entities

• Publication practice for research software:
  • Publication, formal publication
  • Unique identification
  • Incentives

• Metadata:
  • Provision, completeness, correctness, interoperability
Instantiating research software citation graphs: solutions

- Cultural challenges:
  - Software as a research product (Importance principle) – **Policy changes**
  - Practice of software citation – **CIA (Cite It Already!)**
  - Unique identification of individuals and groups/entities – **ORCID**

- Publication practice for research software:
  - Publication; formal publication – **GitHub-Zenodo, Software journals, Software Heritage; new roles for software journals? Business models?**
  - Unique identification – **DOIs**
  - Incentives – **Policy changes, evaluation practices**

- Metadata:
  - Provision, completeness, correctness, interoperability – **Citation File Format (CFF) [6], CodeMeta [7]**
The role of RSEs

- **Cite It Already!** and lead by example

- **Provide citation metadata** in a CITATION.cff or codemeta.json file, help us make CFF better, build tooling to support conversion from CFF to CodeMeta

- **Publish your software** with a DOI

- **Tell your colleagues**, adapt peer reviewing practices to check for software citation
Thank you!


Get in touch!
- stephan.druskat@dlr.de
- @stdruskat
- the coffee queue

References


