

Dare to be mair than just FAIR!

FAIR PLAY to research software,
courtesy of RSEs

FAIR Software @ National eScience Symposium 2019,
Johan Cruijff ArenA, Amsterdam, 2019-11-21

Stephan Druskat |  <https://orcid.org/0000-0003-4925-7248>

(German Aerospace Center (DLR), Friedrich Schiller University Jena,
Humboldt-Universität zu Berlin, SSI Special Collaborator, Board member de-RSE e.V.)



menti.com 71 20 99



seit 1558



Is FAIR software fair enough?

Spoiler: It's not.

But: There's something great about FAIR!

FAIR <insert digital artifact class here>
BETTER RESEARCH

FAIR <insert digital artifact class here> BETTER RESEARCH

- FAIR makes research possible, easier, and better

FAIR <insert digital artifact class here> BETTER RESEARCH

- FAIR makes research possible, easier, and better
- FAIR doesn't obstruct Open Science
(unlike some publishers)

FAIR, the brand

- FAIR, FAIR, everywhere
(it's almost as if ...)

FAIR, the brand

- FAIR, FAIR, everywhere
(it's almost as if ...)



← FAIRY

FAIR, the brand

- We, the research software community, can make use of the brand, e.g.:
 - Policy changes create leverage (to campaign for RSE, or FAIRSE)
 - May increase visibility for software sustainability concerns



Is FAIR enough?

Probably not! [1], [2], [3]

Probably not! [1], [2], [3]

FAIR \neq OPEN

Probably not! [1], [2], [3]

FAIR \neq OPEN

FAIR \neq CREDIT

Probably not! [1], [2], [3]

FAIR \neq OPEN

FAIR \neq CREDIT

FAIR \neq GOOD

Perhaps we don't need FAIR Software, but rather
rules of FAIR PLAY for software!

FAIR PLAY?

(I'm sure there's a backronym in there somewhere)

Perhaps we don't need FAIR Software, but rather rules of FAIR PLAY for software!

FAIR PLAY?

(I'm sure there's a backronym in there somewhere)

FAIR PLAY Software is ...

Findable, Accessible, Interoperable, Reusable,

Perhaps we don't need FAIR Software, but rather rules of FAIR PLAY for software!

FAIR PLAY?

(I'm sure there's a backronym in there somewhere)

FAIR PLAY Software is ...

Findable, Accessible, Interoperable, Reusable,
and Per default open source,

Perhaps we don't need FAIR Software, but rather rules of FAIR PLAY for software!

FAIR PLAY?

(I'm sure there's a backronym in there somewhere)

FAIR PLAY Software is ...

Findable, Accessible, Interoperable, Reusable,
and Per default open source, Licensed,

Perhaps we don't need FAIR Software, but rather rules of FAIR PLAY for software!

FAIR PLAY?

(I'm sure there's a backronym in there somewhere)

FAIR PLAY Software is ...

Findable, Accessible, Interoperable, Reusable,
and Per default open source, Licensed, Attributing
contributions to it (for credit),

Perhaps we don't need FAIR Software, but rather rules of FAIR PLAY for software!

FAIR PLAY?

(I'm sure there's a backronym in there somewhere)

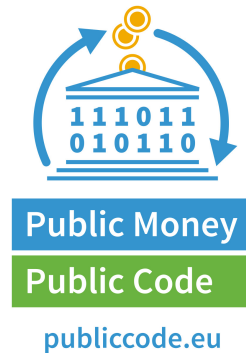
FAIR PLAY Software is ...

Findable, Accessible, Interoperable, Reusable,
and Per default open source, Licensed, Attributing
contributions to it (for credit), and of adequate quality!

$$\text{FAIR PLAY} == \text{FAIR} + \text{RSEng}$$

RSEs be FAIR to the world

- Publicly funded research products (software) [belong to the public](#)
- Use (permissive) open source licensing
- Publish your software proactively in a registry, get a PID (DOI)
- Archive your software sustainably ([Software Heritage](#))



RSEs be FAIR to others

- Do your best to ensure that your software is of “adequate quality”, so that downstream users can:
 1. understand it,
 2. use it,
 3. trust it.
 4. change it.
- How do we measure software quality?
(Internal) metrics? Peer review? Usage?
- Best practices: VCS, CI, tests, documentation, code review, etc.

RSEs be FAIR to themselves

- Make your software citable (“A” in “PLAY”)
- Provide citation metadata

Okay, also be FAIR to others (again)

- Cite software that you use
- Do so both in your papers and in your software

“Haud on a b***in’ minute here, Joe!
Have ye noticed somethin?”



Is this - or is this no - a STUDENTY PLACE?

“Haud on a b***in’ minute here, Joe!
Have ye noticed somethin?”

Q: So I, RSE, am supposed to do all the hard and tedious (metadata) work then?

A: Yes. (But we are trying to make it simpler.)

Citation metadata 101

CITATION files

- Citation File Format (CFF, *YAML*) [4]: citation-file-format.github.io
- (Alternatively CodeMeta (*JSON-LD*) [5]: codemeta.github.io)

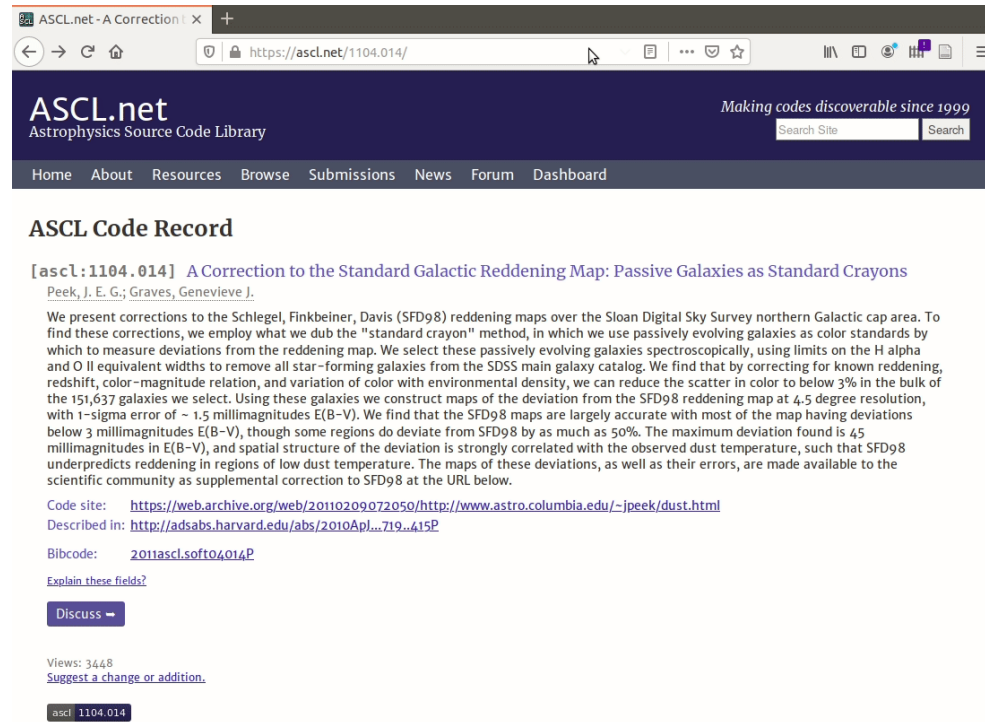
```
# CITATION.cff
cff-version: 1.1.0
message: "If you use this software, cite it using these metadata."
authors:
  - family-names: "Druskat"
    given-names: "Stephan"
    orcid: https://orcid.org/0000-0003-4925-7248
title: "My Research Software"
version: "2.0.4"
doi: "10.5281/zenodo.1234"
date-released: 2017-12-18
```

Manual creation & curation

- Copy, paste & adapt the example, *or*
- use a simple webservice: citation-file-format.github.io/cff-initializer-javascript/.
- Put `CITATION.cff` in the root of your source code repository.

Auto-generation

- Let your build (soon for Maven, incl. dependencies) or registry do it



“But how do I get the Zenodo DOI into the release CITATION.cff file automagically?”

a.k.a.  

We’re working on it. With Zenodo !

And perhaps one day ...



The screenshot shows the GitHub repository page for 'hexatomic'. At the top, there is a navigation bar with tabs for 'Code', 'Issues 34', 'Pull requests 2', 'Projects 3', 'Security', and 'Insights'. Below this, the repository description reads: 'Hexatomic is an extensible software for deep multi-layer annotation of linguistic corpora' followed by the URL 'https://hexatomic.github.io'. A statistics bar displays '361 commits', '9 branches', '0 packages', '4 releases', '2 contributors', and 'Apache-2.0' license. At the bottom, there are buttons for 'Branch: develop', 'New pull request', 'Find file', and 'Clone or download'.

<> Code Issues 34 Pull requests 2 Projects 3 Security Insights

Hexatomic is an extensible software for deep multi-layer annotation of linguistic corpora <https://hexatomic.github.io>

361 commits 9 branches 0 packages 4 releases 2 contributors Apache-2.0

Branch: **develop** New pull request Find file Clone or download

Conclusion

FAIR Software



← FAIRLY

FAIR Software + practice of FAIR PLAY



Conclusion proper

Conclusion proper

- FAIR does not work fully for software

Conclusion proper

- FAIR does not work fully for software
- FAIR is an established “brand”

Conclusion proper

- FAIR does not work fully for software
- FAIR is an established “brand”
- Research software can leverage the brand

Conclusion proper

- FAIR does not work fully for software
- FAIR is an established “brand”
- Research software can leverage the brand
- Even if we fail to define FAIR Software,
FAIR + RSE = FAIR PLAY can achieve for software
what FAIR has achieved for data

Thanks!

citation-file-format.github.io

github.com/citation-file-format/citation-file-format

stephan.druskat@dlr.de




Twitter: [@stdruskat](https://twitter.com/stdruskat)

ORCiD [0000-0003-4925-7248](https://orcid.org/0000-0003-4925-7248)

Slides: [doi:10.6084/m9.figshare.10565717](https://doi.org/10.6084/m9.figshare.10565717)

Appendix

Image credits

- DFG Kodex cover, assumed © by [Deutsche Forschungsgemeinschaft](#) (under fair use)
- Burnistoun animated GIF from [BuzzFeed](#), assumed © by BBC (under fair use)
- Elsevier logo in the public domain 
- Public Money Public Code logo by Free Software Foundation Europe 
- Software Heritage logo courtesy of Software Heritage
- The Open Source Initiative Keyhole Logo by The Open Source Initiative 

FAIRy “art” – use at your own risk

- FAIRy by me 
- FAIR PLAIRy also by me 

References

- [1] Daniel S. Katz, “FAIR is not fair enough,” *Daniel S. Katz’s blog*. Jun-2017 [Online]. Available: <https://web.archive.org/web/20191120175321/https://danielskatzblog.wordpress.com/2017/06/22/fair-is-not-fair-enough/>.
- [2] Daniel S. Katz, “FAIR is not Fair Enough, Particularly for Software Citation, Availability, or Quality.” 2018 [Online]. Available: https://web.archive.org/web/20191120175542/https://www.slideshare.net/danielskatz/fair-is-not-fair-enough-particularly-for-software-citation-availability-or-quality?from_action=save.
- [3] Anna-Lena Lamprecht, Leyla Garcia, Mateusz Kuzak, Carlos Martinez, Ricardo Arcila, Eva Martin Del Pico, Victoria Dominguez Del Angel, Stephanie van de Sandt, Jon Ison, Paula Andrea Martinez, Peter McQuilton, Alfonso Valencia, Jennifer Harrow, Fotis Psomopoulos, Josep Ll Gelpi, Neil Chue Hong, Carole Goble, and Salvador Capella-Gutierrez, “Towards FAIR principles for research software,” *Data Science*, vol. Preprint, no. Preprint, pp. 1–23, Jan. 2019 [Online]. Available: <https://doi.org/10.3233/DS-190026>.
- [4] Stephan Druskat, Jurriaan H. Spaaks, Neil Chue Hong, Robert Haines, and James Baker, “Citation File Format (CFF),” Aug. 2018 [Online]. Available: <https://doi.org/10.5281/zenodo.1003149>.
- [5] Matthew B. Jones, Carl Boettiger, Abby Cabunoc Mayes, Arfon Smith, Peter Slaughter, Kyle Niemeyer, Yolanda Gil, Martin Fenner, Krzysztof Nowak, Mark Hahnel, Luke Coy, Alice Allen, Mercè Crosas, Ashley Sands, Neil Chue Hong, Patricia Cruse, Dan Katz, and Carole Goble, *CodeMeta: An exchange schema for software metadata. Version 2.0*. 2017 [Online]. Available: <https://doi.org/10.5063/schema/codemeta-2.0>.