FAIR principles for research software

Software Source Code IG

research data sharing without barriers
rd-alliance.org

18th March 2020 - RDA 15th Plenary Meeting - Day 1 Melbourne Australia
Contributors

- Michelle Barker, Research Software Alliance (ReSA), Australia
- Salvador Capella-Gutierrez, Barcelona Supercomputing Center (BSC), ELIXIR, Spain
- Neil Chue Hong, Software Sustainability Institute / University of Edinburgh, UK
- Martin Fenner, DataCite, Germany
- Leyla Garcia, ZB MED Information Centre for Life Sciences, Germany
- Carole Goble, Software Sustainability Institute UK / University of Manchester / ELIXIR, UK

continues...

Join our collaborative notes and add your name: http://tiny.cc/FAIR4ResSof
Contributors

- Morane Gruenpeter, Inria, Software Heritage, France
- Daniel S. Katz, University of Illinois at Urbana-Champaign, USA
- Mateusz Kuzak, eScience Center, The Netherlands
- Anna-Lena Lamprecht, Utrecht University, The Netherlands
- Paula Martinez, National Imaging Facility (NIF), The University of Queensland (UQ), Australia
- Fotis Psomopoulos, Institute of Applied Biosciences (INAB), Centre for Research and Technology Hellas (CERTH), Greece
- Mingfang Wu, Australian Research Data Commons, Australia

Join our collaborative notes and add your name: http://tiny.cc/FAIR4ResSof
1. Software Source Code Interest Group (SSC IG) update
2. Updates on related projects FAIR for Research Software, Versioning and Citation
3. Activity - split into 4 different groups for discussions (depending on number of people on call):
4. Discussion
   a. Be part of this new group, you are needed.
   b. Why do we care about FAIR for research software?
   c. What should the scope for a FAIR for research software WG be?
   d. Moving forward with a FAIR4RS WG across RDA, ReSA and FORCE11
Introduction
The Software Source Code IG

Co-chairs:

- Neil Chue Hong
- Julia Collins
- Roberto Di Cosmo
- Mingfang Wu

Objectives:

A forum for discussing research software inside RDA

https://www.rd-alliance.org/groups/software-source-code-ig

Chronology:

RDA P10, Montreal 9/2017 motivations, survey of ontologies, metadata use cases

RDA P11, Berlin 3/2018 identification of gaps in metadata


FORCE11-2019, Edinburgh 10/2019 full day hackathon on Research Software
Summary from P13 session: How to make RS FAIR

<table>
<thead>
<tr>
<th>Findable</th>
<th>identifier/identification, citation, good documentation, metadata (controlled vocabulary, licence, language, contributor), repository for software</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accessible</td>
<td>stable, incentive, reward, licence, citation, access libraries and packages</td>
</tr>
<tr>
<td>Interoperable</td>
<td>apply to software metadata (vocabulary, dependency, operating environment), compatible b/w different versions, archived in containers</td>
</tr>
<tr>
<td>Reuse</td>
<td>dependency, description, documentation, metadata, license</td>
</tr>
<tr>
<td>Other properties</td>
<td>Trusted, Sustainable, Preservable</td>
</tr>
</tbody>
</table>
Definition: Research Software

Research Software is defined by (Hettrick et al., 2014) as:

“software that is used to generate, process or analyse results that you intend to appear in a publication (either in a journal, conference paper, monograph, book or thesis)”

Software plays a triple role in research, as stated in (The Committee for Open Science’s Free Software and Open Source Project Group, 2019):

- It is a tool
- It is a research result
- And it is a research object
Research Software in Open Science

**FAIR** doesn’t necessarily mean *open*, but as stated in ([Mons et al., 2017](#)) requires clarity and transparency to **A**ccess and **R**euse conditions.

“Based on the FAIR software metrics/indicators, communities will be able to agree on degrees of **FAIRness** that the different kinds of software should comply to, in order to reflect their Open Science ideals.” ([Lamprecht et al., 2019](#)).
Update community on efforts around FAIR for Research Software
FORCE11 Software Citation Implementation Working Group
(co-chairs: N. Chue Hong, M. Fenner, D. S. Katz)

Following-on from FORCE11 Software Citation Working Group and the Software Citation Principles it developed

Objective: Produce concrete guidelines for software citation, and implement them within the scholarly research community (software developers, repositories and registries, journals and conference and publishers, indexers, institutions)

A community with monthly calls to discuss challenges and progress in implementing software citation, with task forces for

- **CodeMeta** - standardizing metadata for software, moving towards merging into schema.org
- **Guidance** - developing documents for developers, authors, and reviewer
- **Journals** - coordinating editors and publishers to simplify and implement guidance
- **Repositories** - developing best practices document for handling software
Related Projects - CodeMeta

CodeMeta initiative

- A subset of schema.org
- An academic community discussing software metadata
- A crosswalk table - mapping the metadata landscape

The CodeMeta Task Force

- Part of the FORCE11 SCIWG
- Goal: express all codemeta properties using schema.org

Taken from RDA P11 SSC IG: https://www.rd-alliance.org/sites/default/files/2018-03-22-RDA-IG_0.pdf
Software Source Code Identification WG

Joint RDA & FORCE11 WG which spawned from RDA’s Software Source Code IG & FORCE11’s SCIWG to address this common challenge: how to identify source code

Objectives:

- bring together people involved/interested in software identification
- produce concrete recommendations for the academic community

Remote meeting will be announced in the following days
Related Projects - ReSA

Research Software Alliance (ReSA)

Vision: Research software: recognised and valued as a fundamental and vital component of research worldwide

Mission: To bring research software communities together to collaborate on the advancement of research software.

Task Forces:
- Software landscape analysis
- Evidence for the importance of research software
- Register of research software funding opportunities
FAIR software initiatives and outcomes

“Applying FAIR Principles to Software” at the 2017 Workshop on Sustainable Software Sustainability (WOSSS17)

“Sharing Your Software – What is FAIR?” at the 2018 American Geophysical Union (AGU) Fall Meeting

“FAIR Software” Birds of a Feather meeting at deRSE 2019

Towards FAIR principles for research software 2019 DOI: 10.3233/DS-190026

From FAIR research data toward FAIR and open research software

“Making Software FAIR” at the DTL Communities@Work 2018 Conference

Top 10 FAIR Data & Software Global Sprint, including “10 easy things to make your software FAIR” 2019

“Five recommendations for FAIR software” at NL-RSE 2019

FAIR Computational Workflows 2020 DOI: 10.1162/dint_a_00033

Lorentz Workshop 9-13 March 2020 (Automated Workflow Composition in the Life Sciences)

“FAIRness assessment for software” at the 2018 DBCLS/NBDC BioHackathon

“FAIR principles for Software” at 2019 Workshop on Sustainable Software Sustainability (WOSSS19)

TIB Training workshops on FAIR Data and Software 2018 - 2019

FAIRsFAIR T2.4: FAIR assessment for research software

BRDI NAS Washington 16-17 March 2020

https://www.rd-alliance.org/fair-principles-research-software
https://www.rd-alliance.org/group/software-source-code-ig/wiki/fair4software-reading-materials
FAIR for Research Software
Motivation

- **FAIR principles**
  - Basis for reproducible open science → but more beyond FAIR is needed here
  - Mainly addressing data but aimed to any sort of digital objects → software is a digital object
  - **BUT**

- **Research software is not (just) data** (Katz et al., 2016; Lamprecht et al., 2019)

<table>
<thead>
<tr>
<th>Similarities</th>
<th>In between</th>
<th>Differences</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Not commonly cited</td>
<td>- Can be built on top of (but dependencies on software are more complex)</td>
<td>- Software is more volatile and quickly changing, it is “alive” (evolves and requires maintenance)</td>
</tr>
<tr>
<td>- All FAIR metadata principles apply</td>
<td>- Depend on hardware and software (data for display and production, software for that + execution)</td>
<td>- Reuse comes in different flavors (re-run/execute, reuse, repeat, reproduce, extend)</td>
</tr>
<tr>
<td>- Multiple versions can exist</td>
<td>- Licenses are different (data is not a creative work, software is)</td>
<td>- Can be connected via workflows</td>
</tr>
</tbody>
</table>

- **FAIR principles**
  - Basis for reproducible open science → but more beyond FAIR is needed here
  - Mainly addressing data but aimed to any sort of digital objects → software is a digital object

- **Research software is not (just) data** (Katz et al., 2016; Lamprecht et al., 2019)
### Findability

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>F1</td>
<td>(Meta)data are assigned a globally unique and persistent identifier.</td>
</tr>
<tr>
<td>F2</td>
<td>Data are described with rich metadata.</td>
</tr>
<tr>
<td>F3</td>
<td>Metadata clearly and explicitly include the identifier of the data it describes.</td>
</tr>
<tr>
<td>F4</td>
<td>(Meta)data are registered or indexed in a searchable resource.</td>
</tr>
</tbody>
</table>

### Accessibility

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>(Meta)data are retrievable by their identifier using a standardized communications protocol.</td>
</tr>
<tr>
<td>A1.1</td>
<td>The protocol is open, free, and universally implementable.</td>
</tr>
<tr>
<td>A1.2</td>
<td>The protocol allows for an authentication and authorization procedure, where necessary.</td>
</tr>
<tr>
<td>A2</td>
<td>Metadata are accessible, even when the data are no longer available.</td>
</tr>
</tbody>
</table>

- We have versions and releases
- We have specialized registries
<table>
<thead>
<tr>
<th>Interoperability</th>
<th>Reusability</th>
</tr>
</thead>
<tbody>
<tr>
<td>I1 (Meta)data use a formal, accessible, shared, and broadly applicable language for knowledge representation.</td>
<td>R1 (Meta)data are richly described with a plurality of accurate and relevant attributes.</td>
</tr>
<tr>
<td>I2 (Meta)data use vocabularies that follow FAIR principles.</td>
<td>R1.1 (Meta)data are released with a clear and accessible data usage license.</td>
</tr>
<tr>
<td>I3 (Meta)data include qualified references to other (meta)data.</td>
<td>R1.2 (Meta)data are associated with detailed provenance.</td>
</tr>
</tbody>
</table>

- Vocabularies work well for software metadata
- Metadata should target machine readability and data exchange
- Software product, i.e., data should be FAIRly described
- And still, what is an interoperable software? Are we talking about workflows, containers or so?
- Software dependencies and their licenses should be covered
- We also need software documentation (including use case examples)
- Re-run, reuse, repeat, what are we talking about?
Discussion
Software is not data

ice breaker activity, 5 minutes to share your thoughts
Let’s start

Full room discussion or in groups depending on how many people. 25’ and 10’ wrap up

- Introduce yourself to your neighbours (name, affiliation, why do research software interest you?)
- How can the FAIR principles be applicable to software?
  - Should we apply them? Why? What strategies?
  - What can this community do to improve FAIRness for research software?
- Parallel discussion:

<table>
<thead>
<tr>
<th>Metadata</th>
<th>Indicators/ Metrics</th>
<th>Versioning</th>
<th>FAIR scope</th>
</tr>
</thead>
<tbody>
<tr>
<td>What controlled vocabularies are currently available to describe software and its associated metadata?</td>
<td>What aspects should be taken into account in order to create indicators to measure FAIRness in software?</td>
<td>Software versioning and how to handle releases? A discussion about software provenance.</td>
<td>Why do we care about FAIR for software? What is the higher-level goal of FAIR? Is FAIR by itself going to help us get there?</td>
</tr>
</tbody>
</table>

Why FAIR for Research Software?

- Do we need a Working Group on FAIR for Research Software?
- What would the scope of the WG be?
  - Source code, Executables, Containers, All.
  - Domain specific, Scientific, All research software
  - What is out of scope?
- What other RDA WGs or IGs should we collaborate or connect to?
- What kind of outputs can this WG produce?
  - Recommendations? Guidance?
  - Assessment tool?

Let’s work collaboratively on the notes: [http://tiny.cc/FAIR4ResSof](http://tiny.cc/FAIR4ResSof)
Collecting information

Let’s work collaboratively on the notes: [http://tiny.cc/FAIR4ResSof](http://tiny.cc/FAIR4ResSof)

- Names, institutions, emails, level of involvement (1-3 High to low)
- Next steps
- What else should be added to the Wiki of materials?
Task force aims

- Agree what's in scope for FAIR for software
- Define FAIR for software - what are the sub-elements of each letter
- Potentially define metrics of how this is measured, or even stages of adoption (minimal, good, best) for each subelement and each letter
- Coordinate community consultation to reach agreement of FAIR for software principles
- Identify implementation approaches for the principles
- Identify funding mechanisms to support this work

What else?
A community taskforce, supported by:

FORCE11
The Future of Research Communications and e-Scholarship

Research Software Alliance


fair-software.eu - endorsed by several organizations

The Software Heritage research software guidelines https://www.softwareheritage.org/save-and-reference-research-software/


Di Cosmo, Roberto; Gruenpeter, Morane; Marmol, Bruno; Monteil, Alain; Romary, Laurent; Sadowska, Jozefina. Curated Archiving of Research Software Artifacts: lessons learned from the French open archive (HAL) Slides from IDCC 2020 presentation at https://doi.org/10.5281/zenodo.3667713, full article https://hal.archives-ouvertes.fr/hal-02475835.
Thanks for joining