Copy of the notes from RDA VP16

Breakout 3: Software Source Code IG

Tuesday, 10 November 2020/5:00 - 6:30 PM UTC

Existing efforts and practices related to Software Source Code in academia

Useful links

Session page on RDA VP16 programme

slides

SSC IG group page

Meeting objectives

1. Software updates
2. Collecting practices related to software source code in Academia

Agenda

Introduction : Interest Group goals and past activities **Ice-breaker**: Why are you interested in software source code?	10
Software source Code IDentification (SCID) WG output: Use cases and identifiers schemes for software source code identification	15
FAIR4 Research Software WG: Introduction and Invite to join the discussion	5
FORCE11 Software Citation Implementation Working Group update ● including the ongoing task forces (CodeMeta, journals, repositories...)	10
Overview of other ongoing efforts related to software	10
Group activity: Collecting existing practices	30
Next steps for the SSC Interest Group	10

Participants

	Name	***Institution***	***country***
1	Morane Gruenpeter	Inria & Software Heritage	France
2	Daniel S. Katz	University of Illinois	USA
3	Christian Pagé	CERFACS	France
4	Carlo Zwölf	Paris Observatory / VAMDC	France
5	Matt Cannon	Taylor & Francis	UK
6	Julia Collins	NSIDC/CIRES/CU	USA
7	Becca Wilson	University of Liverpool	UK
8	Amy Nurnberger	Massachusetts Institute of Technology	USA
9	Rossella Aversa	KIT	Germany
10	Viviana Letizia	SoftwareX	France
11	Ville Tenhunen	EGI Foundation	The Netherlands
12	Josh Greenberg	Alfred P. Sloan Foundation	USA
13	Gerrit Günther	Helmholtz-Zentrum Berlin	Germany
14	Robert Ulrich	KIT	Germany
15	Leighton Christiansen	National Transportation Library, US Dept of Transportation	United States
16	Wolmar Nyberg Åkerström	NBIS - National Bioinformatics Infrastructure Sweden / Uppsala University	Sweden
17	Neil Chue Hong	SSI / University of Edinburgh	United Kingdom
18	Fernando Aguilar	CSIC	Spain
19	Jonathan Petters	Virginia Tech	USA
20	Thu-Mai Christian	Odum Institute, University of NC at Chapel Hill	US
21	Andreas Rauber	TU Wien	AT

22	Hannes Thiemann	DKRZ	Germany

Ice-Breaker

Why are you interested in Software Source Code?

	Name
I want to help ensure that people who develop and maintain code get credit for it	Dan Katz
As a developer myself and in a research institute (numerical modelling) associated with universities, we have many people writing source code: PhD students, Post-Docs, Trainee Students, Engineers, Researchers. Many source codes are hosted on gitlab (internal server), public gitlab and github, or also without any revision system. I am interested in having some information on what would be recommended on how to have and implement best practices in writing software.	Christian Pagé
To hear about latest activities in software, to ensure our journals can give accurate advice, support and functionality to promote citing and linking of software to research papers	Matt Cannon
For making research reproducible/repeatable it's essential that code used is make accessible and usable, and that researchers generating the code get credit for the work involved	Jonathan Petters
Working on open source projects, interested in ensuring credit for software and methodology development	Becca Wilson
To improve Research Software quality, adopting FAIR principles and giving value as a research product.	Fernando Aguilar
Software codes are one essential part of the reproducibility of science and it also needs support, resources and governance (to be FAIR)	Ville Tenhunen
Our organization releases code to use in analysing datasets we manage. I want to allow people to cite the software as well as citing our data.	Julia Collins
For research to be reproducible	Viviana Letizia

Reproducible research and transparent research	Becca Wilson
The updated US DOT Public Access plan will include Software and Code, along with reports and datasets, as research outputs that must be managed for sharing and preservation. Software and code will need to be shared with the public.	Leighton Christiansen
I want to make it easy for researchers and research engineers to develop, collaborate on and maintain high integrity research software without being computer science specialists.	Wolmar Nyberg Åkerström
I work at a library and manage a project to help organize, reference and preserve code of scientific software. Besides that I code myself.	Robert Ulrich
We share data produced by theoretical codes. For sharing data through web-services, we build ad hoc code. All our activity is based on code production.	Carlo
Reproducible and transparent research. To be compliant with FAIR best practices in the data management plans of European projects	Rossella Aversa
See newly framed Sloan program Better Software for Science	Josh Greenberg
To ensure reproducibility of reported findings	Thu-Mai Christian
I want to help people improve the maintainability and reusability of their source code	Neil Chue Hong
As the person responsible for a certified data archive I am interested to learn if and how the principles of data management can be transferred to software.	Hannes Thiemann

Notes

Please help us write collaborative notes from here, this document will be used to collect the updates summary and group discussions. Add headings 2 or 3 whenever possible.

Group activity: Collecting existing practices

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

- Introduce yourself to your neighbours (name, affiliation)
- Software practices collection:

- Do you or your organization create software? Use software?
- Do you or your organization follow institutional or community best practices with the source code you create? (an old (2020) example is the Software Release Practice by E.S Raymond)

All

Do you or your organization create software? Use software?	**Do you or your organization follow institutional or community best practices with the source code you create? (links are welcome, but you can also describe the practice)**	**contributor**
Yes and yes	Actively aim to produce reusable pipelines where possible and contribute packages to repositories such as Bioconda. My institution teaches internal and external workshops on reproducible data analytics using version control, automated builds, dependency management, and containers.	Wolmar Nyberg Åkerström
Yes and yes	It depends on the software. We have internal training, and follow SSI best practices for software products we create to ensure they have licenses, use version control and publish specific versions with identifiers. This is less formal for one-off scripts.	Neil Chue Hong
Yes and yes (archiving code)	Probably not, in archiving...it'd be good to have a short list of high impact/low energy actions to take in archiving source code	Jonathan Petters
Yes and yes.	We create software as companion artifacts to some of our data collections. We also create software for e.g. web applications that impacts data acquisition (data may be sliced/reprojected), so it's important for data provenance to represent what's happening. Depending on who's creating the software, we try to follow best practices in terms of release tagging, versioning, and testing. Often, though, code developed for a specific dataset is not created by a "professional software developer," so the same best practices may not apply.	Julia Collins
Yes and Yes	Git-based tools (institutional GitLab),	Rossella

	documentation, versioning	Aversa
Yes and Yes	I'd need to check but i think we treat most software we create as proprietary (this could be around copyediting or typesetting articles; as part of article submission systems etc)	Matt Cannon
Yes and Yes	Internal use of institutional git, Git Hub for collaborations with people out of our institute and GitHub/Zenodo integration for citation of codes in papers and other works (like data)	Carlo Zwölf
Yes and yes	Usually we use good enough / best practices we find from others for open community software - sometimes using guidance from BSSw / ELIXIR, and sometimes based on discussions in RSE groups (or SORSE) and NumFOCUS. We also use (and teach) Software Carpentries material.	Dan Katz
Yes and Yes	My organisation leaves research groups to do this themselves. We develop open source software, everything is maintained in public facing version control, integrated with Zenodo. Currently working our way through deposit of software in CRAN repository. Generally follow guidance from the SSI. Spent huge amounts of time on linting, unit and integration testing.	Becca Wilson
Yes and Yes	It highly depends on who writes the software. Most of them are using gitlab or github. But a significant part of the software is also developed in "research" mode, and can be reused and modified over several years without being really organized in a more standard way. But git is getting more widespread even among less technical people. Software citation is important and is often overlooked in scientific publications, or not done at all.	Christian Pagé
Yes and yes	Mostly orally transmitted best practices, unfortunately, in technical meetings. However we do distinguish 2 types of software: personal, one-shot code and distributable code which needs much higher standards (documentation, unit tests, …) Software citation for us is used in two typical cases: for software we use (e.g. library x in	Fernando NIño

	version y.z) and to cite good practices or bugs (lines m-n of file balbla.py were intended to do this, but they really do something else, not what developers expected …).	

Next steps questions

- What subjects would you like to discuss during the next plenaries?

Subjects	+upvoters
How to describe software with software metadata	+6
Onboarding new users (in academia) to version control	+4
Code quality assessment - quality of algorithms and the form of code.	+7
Community curated repositories for trusted software artifacts (e.g. BioConda)	+2
Basic ”literacy” on dependency management and risk assessment when using packages (e.g. know that you're executing arbitrary code on you computer and what that means)	+3

- What types of materials would be helpful to have on the SSC IG wiki page?
  For example we have previously added materials here: https://www.rd-alliance.org/group/software-source-code-ig/wiki/fair4software-reading-materials

Materials	Contributor +upvoters
Lists of best practices for code development (source code version control, continuous integration, repositories and citation, community involvement, etc.)	+5
Links to plenary session Google Docs (notes, slides)	+4
Success stories	+1

Links to achievements (papers, recommendations, etc.)	+3
Links to ‘FAIR’ exemplars of shared source code	
Metadata schemas for software	+1

- Would you like the **mailing list updates** to be more frequent and if so, what are the topics you would like to see on the mailing list?

Topics	Contributor +upvoters
We are over informed by several RDA mailing lists + others… it is hard to process in detail all the information. Low frequency rate is suitable.	+1
Experiences on implementing/adopting best-practice, e.g. showcases, adoption stories	+3
conferences/seminars/training	+3
Can the wiki be configured to send updates when information is added there? That would be useful to remind me to view the information.	
Maybe information can be summed up to a regular mail, not to overwhelm	
Updates you’ve been sending that summarize upcoming plenaries and reminders of working group meetings are helpful.	

Feedback

Thanks for joining us!!!

Let us know your thoughts of this session, we are looking to improve (please write here in the document or email morane@softwareheritage.org)

Chat transcription

17:53:03 From Morane Gruenpeter : Collaborative notes https://tinyurl.com/y2kunpf5 These slides https://tinyurl.com/yyargmeu
17:58:17 From Daniel S Katz : Collaborative notes https://tinyurl.com/y2kunpf5 These slides https://tinyurl.com/yyargmeu
17:58:39 From Daniel S Katz : Please sign in
17:58:46 From Daniel S Katz : in the notes

18:10:53 From Josh Greenberg : I'd add that "open source" is as much if not more about the practices around the code (collaborative production/maintenance/etc) as a licensing choice.
18:11:51 From Amy Nurnberger (she/her) [MIT] : +1 , and good documentation!
18:41:55 From Daniel S Katz : https://www.researchsoft.org
18:43:50 From Daniel S Katz : to vote, click on participants on the bottom of zoom, then you will see yes and no below the list of participants in the new pane
18:44:13 From Josh Greenberg : I apologize but I have to hop off to another meeting in ~10 mins, so will abstain from voting :)
18:44:15 From fernando.nino@legos.obs-mip.fr To Morane Gruenpeter(Privately) : CAn you please repeat the question ?
18:44:47 From Wolmar Nyberg Åkerström : Discuss in large group: Yes
18:45:08 From Wolmar Nyberg Åkerström : Ah, I misunderstood. ^^;
18:52:00 From Josh Greenberg : One could imagine as a thought experiment treating some source code in the same way we treat private data; wrapping it in privacy-preserving systems like differential privacy. Not clear I can come up with a use case.
18:52:35 From Josh Greenberg : The Hathi Trust takes this approach for copyright-encumbered works, allowing "non-consumptive" algorithmic textual analysis
18:53:04 From Neil Chue Hong (he/his) : Here, I'm wondering if you have no access to the source code, can you do open science. Agree that there's a spectrum of openness.
18:53:24 From Daniel S Katz : private data can also have that aspect, that the data can be viewed under some limited agreement
18:53:55 From Amy Nurnberger (she/her) [MIT] : @Neil this is an interesting question, especially as more AI/ML/neural nets, etc are applied to research problems
18:53:57 From Josh Greenberg : @Dan yes, and that access can be regulated through policy gatekeeping or technology, or both
18:54:34 From Neil Chue Hong (he/his) : @Amy - I've just been on a panel at the Open Data Institute about algorithmic transparency, which is what prompted my question.
18:55:17 From Amy Nurnberger (she/her) [MIT] : From Christian's comment, I think having code that you can re-run calls into question whether or not it supports science, let alone open science
18:56:00 From Amy Nurnberger (she/her) [MIT] : *can't
18:56:13 From Josh Greenberg : Have been thinking lately about how the question of "source code" gets complicated by a trained neural net, which is not interpretable in the way that

source code is. There's a lot of activity right now in the "AI transparency" world, that I'm not sure how it reconciles with the agenda here.
18:56:28 From Neil Chue Hong (he/his) : @Wolmar - I think having good tools (like reference managers that understand software) will be a key to adoption of better software practices. There are a number of open source and commercial ones that do have some support, but it could still be easier. I want the equivalent of the button that just identifies the software and clips the reference for me.
18:57:13 From Daniel S Katz : A poster I'm presenting (31b) is on FAIR for ML models - these models are between data and software, and have aspects of both
18:57:27 From Neil Chue Hong (he/his) : @josh - the ODI panel did generally come round to the position that we have a lot of discussion around AI transparency and not enough on AI assurance.
18:58:00 From Josh Greenberg : @Neil/Wolmar - I have a soft spot for reference management (see: Zotero), and would love to discuss further if anyone has ideas about what a reference manager for software would look like; even a requirements gathering exercise and gap analysis could be useful.
18:58:08 From Daniel S Katz : we would like to start an IG or WG on this topic (FAIR for ML) - if you are interested, please email me
18:58:24 From Josh Greenberg : @Dan - please do loop me in
18:59:04 From Josh Greenberg : (Sorry, gotta run - great discussion!)
18:59:20 From Neil Chue Hong (he/his) : @josh - AI assurance should be similar to testing and documentation for software engineering. The hard part is not developing the software or training the ML model, it's being sure that it does what you chose it to do.
19:02:41 From Julia A Collins : Fernando's comment touches on the issue of trust, and assessing quality of open software. I'm not prepared to take those any further right now, though. :-)
19:02:55 From Julia A Collins : Just some food for thought.
19:06:19 From fernando.nino@legos.obs-mip.fr : Yes, and to doing it well, the reference management software will be necessary, so as to tackle dependencies on other software and maybe have alerts if a bug was found in a particular part of code of a particular library you are using or assessing...
19:06:21 From Jonathan Petters : Think there's also a big difference in releasing source code when it's meant to be a framework/model to share with a community, and when releasing source code to back up one research project…what should be expected in the quality of the code might be different
19:06:30 From Christian Pagé, CERFACS : @Amy most of the code that is used to perform data analysis and processing (long tail of research) are ad-hoc developed scripts in research mode and most of them are very specific on local architecture and not really organized. This is still open science because you share the methodology and algorithm.
19:06:54 From Christian Pagé, CERFACS : @Jonathan yes we have both of these types of source codes in our institution
19:15:03 From Wolmar Nyberg Åkerström : Regarding reference manager for Software I was thinking something on the lines of a software artifact repository manager, e.g. Nexus, with curated metadata and artifacts specifically for research purposes.

19:18:35 From Julia A Collins : Exactly what I was thinking, Neil! +1 for The Carpentries
19:18:41 From Jonathan Petters : https://swcarpentry.github.io/git-novice/
19:26:18 From carlo zwolf : The web site related to Edinburg plenary disappeared. Not sure if it is a bug on the web site, or the consequence of a decision about pandemic…
19:26:29 From carlo zwolf : RDA plenary
19:27:58 From Neil Chue Hong (he/his) : I think it's more a bug because with the start of thes RDA plenary, the "Next" one is P18 but the "Planned" one is still this one (P16)
19:28:07 From Thu-Mai Christian (she/her/hers) : Great discussion, thank you!
19:28:21 From Wolmar Nyberg Åkerström : Great session!
19:28:46 From Julia A Collins : Excellent discussion, thanks!
19:28:46 From Christian Pagé, CERFACS : I really liked the session, thanks!
19:28:51 From Neil Chue Hong (he/his) : Thanks everyone - great chairing, Morane!
19:28:55 From Jonathan Petters : Thanks very much, enjoyed it!
19:28:58 From Daniel S Katz : posters in 90 minutes ...
19:28:59 From Amy Nurnberger (she/her) [MIT] : THank you for the session!
19:29:00 From Robert Ulrich : Thx! Drink coffee and keep coding.