

Software Source Code focus group (BoF)

Program online at:

<https://www.rd-alliance.org/software-source-code-focus-group-rda-9th-plenary-bof-meeting>

This document is: **bit.ly/RDA9Software**

The introductory slides are available at:

<https://annex.softwareheritage.org/public/talks/2017/2017-04-07-RDA-BoF-source-code.pdf>

Index:

- [Meeting Agenda](#)
- [Meeting Objectives](#)
- [Introduction](#)
- [Reference Materials](#)
- [Notes from BoF](#)

Introduction

In order to make progress, Science and Technology need to share and reuse knowledge that is traditionally described in scientific articles, and is often based on data that is collected through experiments or produced by a variety of processes. Infrastructures for collecting, preserving, referencing and sharing articles have existed for a long time, and data is now clearly in the picture.

But a rapidly increasing part of our collective knowledge is embodied in, or dependent on, a wealth of software artifacts, that are now an essential component of all modern scientific and technological development. Hence, modern e-infrastructures that support scientific and technical development must take into account this third important element, software, and in particular the source code of software.

Software Source Code is a unique form of knowledge: executable by a machine, it is at the same time meant to be human readable, as it is the preferred form of work for a developer. Collecting, preserving and providing access to the source code of the software artefact is essential for the reproducibility of scientific results, recording methodology, understanding the fine points of scientific and technical developments, and building on top of existing knowledge.

Reference Materials

- The 100 top papers (Nature, Oct. 2014) show that software is essential for science

<http://www.nature.com/news/the-top-100-papers-1.16224>

- Report on software sustainability
(http://www.ncdd.nl/wp-content/uploads/2016/11/201611_DE_Houdbaar_final_r...)
- The Software Sustainability Institute (<https://www.software.ac.uk/>)
- The Artefact Evaluation initiative in Computer Science
(<http://www.artifact-eval.org/about.html>)
- The Software Heritage initiative (<https://www.softwareheritage.org>)
- Enhancing reproducibility for computational methods
(<http://science.sciencemag.org/content/354/6317/1240.full>)
- Preserving Complex Digital Objects
(<http://www.facetpublishing.co.uk/title.php?id=049580#.WG-heaKLSL4>)
- Software Citation Principles (<https://peerj.com/articles/cs-86/>)
- Guidelines for persistently identifying software using DataCite
(<http://rrr.cs.st-andrews.ac.uk/wp-content/uploads/2015/10/guidelines-sof...>)

Meeting objectives

This Birds of Feather meeting has the goal to bring together inside RDA people and partners interested in charting the existing activities around source code sharing, preservation, and reproducibility; identifying the core issues; and working on an agenda for developing a coordinated set of actions to take properly software source code into account in the scientific arena. We would like to point towards pathways for implementing these actions, through infrastructure, interoperation of infrastructures for software preservations, policies and guidance. As an outcome of the BoF session, we would like to identify whether there is interest and effort to establish an Interest Group, specific Working Groups, or contribute to existing Working Groups.

Meeting agenda

Chairs: Roberto Di Cosmo and Neil Chue Hong

1. Presentation of the idea behind this BoF (Roberto Di Cosmo) - 15 minutes [09:00 - 09:15]
2. Survey of open issues and existing initiatives (discussions, all participants) - 45 minutes [09:15 - 10:00]
3. Summary of the discussion (Neil Chue Hong) – 15 minutes [10:00 - 10:15]
4. Definition of the next steps (Roberto Di Cosmo, Neil Chue Hong – 15 minutes [10:15 - 10:30]

Notes from the BoF Session

- Is scientific software different from any other software?
 - Scientific software depends on much non-scientific software
 - Lots of small “throwaway” codes

- Source code has specific referencing identifier mechanism for storing history (commit hashes)
 - Is this an issue if you move version control repositories or mechanisms?
 - Not in theory as the identifier contains the intrinsic information, can convert
- what about environment, dependencies etc - can most researchers usefully preserve alongside code via Docker etc?
 - To be asked
- Other things that should be preserved alongside
 - Mailing lists, bug trackers etc
- Should you archive every single commits, rather than just versions?
 - It is possible - yes?
 - Is it necessary?
 - Archives should not make a decision for you - if you expose everything it should be stored
- Distinction between software as knowledge vs software as the thing that causes a computation
 - Different from previous discussions
- Archival practice not fully aligned with what was described in slides
 - They throw things out, they choose not to take things
 - Git allows rebasing, which allows erasing of history - want keep changes to commit history as well
- Keeping data attached to publications
 - Archiving data - if we say we want to archive everything, the comeback is that it is too expensive
 - Is software cheaper?
 - Long tail of data
 - Quality check
- IG? Best practice for depositing software in archives
- Other groups
 - Reproducibility IG
- Metadata to improve discovery
- Both software and source code have a meaning
 - In short term, want to reproduce the software
 - In long term, want to understand the history/background
- Analogy with scholarly communication
 - Don't get to read science journals first five drafts
 - Does this discourage publishing
 - Also with literature archives
- Where should this BoF go
 - Source code is easier and better scoped
 - What are the biggest problems we're trying to solve?
 - Archival
 - Discoverability
 - Reproducibility
- What can RDA learn from modern programming practices in version control

- Most people in room interested in archival of source code and discoverability, rather than reproducibility
- RDA has bits of software and data in many groups
 - Most modern data is meaningless without software
- Archiving available code (i.e. not proprietary code) is already a huge win for the user
- Take it forward to reuse
 - Might require things like test data
 - Thinking about why are we doing this might help
- Missed things like Internet Archive in the slides
 - Archiving solutions might be good enough for software and its related artefacts
 - For this discussion, question is whether we need something for scientific software rather than software in general?
 - Scientific software is less well curated in general than software
 - Can overlay ?(scientific archival / discoverability) on top of archival
- Top level seen by scientist is the journal
 - Does discoverability need to have a reference to a publication?
 - There are data formats which allow you to reference the version of software used to create/analyse that file (e.g. crystallography). Might be able to include closed source in this information
 - Archives you use a name (author) as access
- What would be the different parts that we can solve in RDA?
 - Guidelines around metadata
 - To improve deposit in repositories / archives
 - Comparison of examples of how software is handled in different disciplines
 - How handled in practice
 - IG Metadata has discussed software metadata
 - Get expertise into this from software experts
 - Suggested that information summarised in Software group and fed into Metadata group, start from use cases
 - CodeMeta
 - No “software and data” interest group
 - Summarise and disseminate knowledge to rest of RDA
 - Use Cases
 - Metadata efforts in RDA scope are much broader than what we are discussing in this room. Chairs (Keith Jefferey and Rebecca Koskela) would be interested in knowing that there are people wanting to tackle software size. Keep mutually informed.
 - Recommendations for repositories / guidelines for both
 - Source code for reproducibility
 - Source code for knowledge
 - Already have good tools and workflows for archiving software compared to others
 - For RDA would be good to understand relation between the software (archived in one place) and the data (archived in another)

- In PREMIS metadata have created new metadata to link software, data and at high level
- Provenance IG / WG
- Publishing Data Workflows
 - Lot of work on software citation at FORCE11
- Question around RDA governance
 - How would a crosscutting subject like software fit into RDA?
 - Interest group might be the right vehicle
 - Tell other groups that this is going on
- Quality and Sustainability of Software work that is likely to go into FORCE11
 - Is this well distinguished for the scope of this group?
 - People related to both initiatives
 - Can create common groups between organisations
 - Or just communicate and have joint members

Survey of open issues and existing initiatives

Questions for people in the room:

- How many people use software?
- How many people develop software?
- How many people are involved in infrastructure to share or preserve software?
- How many people are involved in creating or disseminating policy / guidelines around software?
- How many people think that we have already solved the challenges of sharing and preserving software?

Discussion points:

- Software: source code vs other forms (binaries, virtual machines)?
- Snapshot/releases vs. development history
 - See also work in RDA on Dynamic Data Versioning
 - Are software versioning use cases different?
- Is research software different from other software?
 - Should we look at/learn from the (free software) developer communities?
 - Do we need specific research software archives, or can we all share a general purpose archive?
- Metadata to aid discovery and retrieval
- Fit with existing RDA and other (e.g FORCE) working groups
 - Reproducibility & Provenance
 - RDA IG Research Data Provenance / Provenance WG
 - Identifiers
 - WG PID Kernel Information
 - Versioning
 - WG Data Citation
 - IG Data Versioning

- Management Plans
 - IG Active Data Management Plans
- Discovery
 - ?
- Is it useful to collect and publish use cases?
 - Who can contribute
- Machine Actionable Software Management Plans

Existing Initiatives:

- Software Description/Citation/Discovery
 - CodeMeta
 - SciCrunch
 - ASCL
 - NSF Workshop on Supporting Scientific Discovery through Norms and Practices for Software and Data Citation and Attribution -
https://softwaredatacitation.org/Workshop%20Report/SoftwareDataCitation_workshop_report_2015_April_20_with_logo.pdf
 - Software Attribution for Geoscience Applications (SAGA) - a software tool for citation of open source software that describes the software environment and attribution for contributions from multiple authors -
<https://geodynamics.org/cig/projects/saga/> FORCE11- Software Citation Implementation WG
<https://www.force11.org/group/software-citation-implementation-working-group>
- Preservation repositories for software
 - Snapshots/Curated
 - Zenodo
 - Figshare
 - Institutional repositories
 - Software Museums (including Internet Archive software collections)
 - Libraries/Archives (remark special case of France legal deposit)
 - Generic Crawling
 - Internet Archive
 - Full fledged version control software knowledge base
 - Software Heritage
- Software Management Plans
 - Using DMPOnline for Software Management Plans:
<https://www.software.ac.uk/software-management-plans>

Next Steps

Should this BoF be converted into a:

- Interest Group?
- Working Group(s)?

Who is willing to contribute effort to develop a charter or case statement?

- An interest group which looks at the relation between software, source code and research data
 - Which collects and publishes use cases of current examples and practices
 - Which identifies and communicates expertise to related other groups which could benefit from expertise
 - Julia Collins
 - Michelle Barker
 - Mingfang Wu
 - Ian Bruno
 - Jonathan Tedds
 - Martin Hammitzsch
 - Morane Gruenpeter (SWH)
- Resolve name of IG, email address and chairs

Background and interests of BoF Proposers

- Neil Chue Hong: via SSI, deep experience with helping scientists improve the overall quality of the software they create in order to enhance reproducibility, so your contribution would be very significant in terms of charting best practices/policies, and suggesting pathways to make them better known/shared
- Patrick Aerts: with DANS, wrote an impressive report on Software Sustainability, and could share some of its conclusions, and contribute, with Neil, to outlining a plan of action on educating developers that could be one of the outcomes of the BoF
- Roberto Di Cosmo: can present the approach to sustainability taken by Software Heritage in building the universal source code archive, a common infrastructure which is just one of the many necessary building blocks for actual software sustainability; drafting guidelines that help software developers (scientific or not) making their source code easily archivable and referenceable could be a contribution to the outcome of the BoF
- Victoria Stodden: an impressive work for raising awareness on the importance of reproducibility, and I'm sure you have precious experience to share on what are the blockers and the objections encountered, and what are the best ways to overcome them; summarizing some of these issues would be a significant contribution to the outcome of the BoF