WG Software Source Code: Identifying, Referencing and Citing Source Code of Research Software

Group page:
https://rd-alliance.org/wg-software-source-code-identification-13th-rda-plenary-meeting

Group Wiki:
https://rd-alliance.org/group/software-source-code-identification-wg/wiki/kickoff-wg-p13-references-software-source-code

Meeting objectives:

This is the first physical meeting of the working group, which is a joint RDA/FORCE11 effort. The objectives of the meeting are:

1) Kickstart the group activity, introducing the group members, objectives and workplan
2) Discuss the different reasons for and objectives of software source code identification, in academia and in industry
3) Document the state-of-the-art, with information on the relevant initiatives and ongoing activities that have been taking place in the area of software source code identification

Meeting agenda:

- Introduction (10m)
    - Software is knowledge
    - Software source code is special

- When trying to get the source code research papers were based on, a 40% failure rate was observed. Software was not findable or exact version of the software was missing (Christian Collberg, https://doi.org/10.1145/2812803)
- Software is a pillar of open science, but was forgotten for far too long
- Interest in research software is raising, academic credit
- Accept the complexity of software, it's not just data
- We must learn from what exists, not reinvent the wheel
- Fragmented landscape, we have academic initiatives, industry initiatives

- Discussion on motivations and difficulties (20m)
  - Dependencies, complexity of the object
  - Credit, citations: the author may not be happy to see the software mentioned in all the different versions, credit spread around all the versions
  - Repos are dependent on commercial providers (GitHub), what about the long term plans
  - software tends to move around, if a platform specific identifier is used it may get lost, so PIDs need a layer of abstraction
  - Identification for credit and identification of software are different things
  - Reproducibility, credit and transparency are motivations
- Conceptual framework for source code identification: DIOs and IDOs (15m) see the iPres 2018 paper available at https://hal.archives-ouvertes.fr/hal-01865790

- Presentation of a few proven approaches (30m)
  - Software Heritage's swh-id, e.g.:
- Apollo 11 Master ignition routine:

SWH-ID identifier
swh:1:cnt:41ddb23118f92d7218099a5e7a990cf58f1d07fa
With the resolver prefix added
https://archive.softwareheritage.org/swh:1:cnt:41ddb23118f92d7218099a5e7a990cf58f1d07fa
With contextual information added
https://archive.softwareheritage.org/swh:1:cnt:41ddb23118f92d7218099a5e7a990cf58f1d07fa;lines=53-72;origin=https://github.com/chrislgarry/Apollo-11/

  - ASCL.net (Astrophysics Source Code Library) [presentation]

- Wrap up: summary of results and next steps (10m)

Attendees (add yourself here)

Name	**organization**	**do you/your organization develop software?**	**do you/your organization use identifiers for software?**	**if yes, which identifiers and for what purpose?**
Roberto Di Cosmo	INRIA/SWH	yes	yes	Swh-id for reproducibility; HAL-id for moderated metadata
Fernando Niño	IRD	yes	yes	DOI & mercurial hash, and Reproducibility and identification of processing chain creating distributed products.
Peter Neish	University of Melbourne	yes		
Pierre Montagano	Code Ocean	Yes	Yes	We containerize executable code, using Docker then assign a DOI
Alice Allen	Astrophysics Source Code Library (ASCL)/UMD		Yes	ASCL ID for all software listed in the ASCL, and DOI for those codes we house
Julia Collins	NSIDC/CIRES /CU	yes	no	
Stephanie van de Sandt	CERN	yes	yes	DOIs (Zenodo)
Amy Hodge	Stanford	no	sometimes	code deposited into

	University			our repository is issued a local unique identifier; we can also assign a DOI if requested
Morane Gruenpeter (remote)	SWH & Crossminer	yes	yes	Swh-id for reproducibility; HAL-id for moderated metadata
Jez Cope				
Tovo Rabemanantsoa	INRA	yes	no	
Kathryn Unsworth	CSIRO	yes	yes	DOIs - publish software in CSIRO's Data Access Portal (DAP)

Session notes:
Introduction to the Software Source Code identification WG
A joint WG which spawned from the RDA's Software Source Code IG and FORCE11's SCIWG
Software is an important pillar of Open Science but it has been forgotten.
Not recognized as a first class citizen.
Lack of guidance/consensus on how to choose a license, cite software, relate to industry best practices and make source code FAIR.
Today an interest in (research) software is rising:

- Artifact evaluation
- Reproducible research
- Software archival
- Academic credit

For all this we need identifiers for research software
Challenges:

- Complexity

- Fragmented landscape

Discussion: motivations & difficulties

- Identifying a software with different versions can be difficult
- Why not use the hash for identification?
- Reuse of the software components and the credit should be attributed

Software projects are not born equal

- Many differences:
  - Structure
  - Lifetime
  - Community
  - Authorship
  - Authority

Conceptual framework to qualify identifiers:
Functions for identifiers:

- Generation
- Assignment
- Retrieval
- Verification
- Reverse Lookup
- Description

What do I need for identifiers that are compatible with reproducibility?

- Integrity
- No middle man

We could not find systems that answer integrity and no middle man.
An important distinction between DIOs and IDOs.
DIO

- (potentially) non digital objects
- Epistemic complexity
- Need an authority

IDO

- Only digital objects
- Can provide both integrity and no middle man
- Broadly used in modern software (git, etc..)

IDOs are enough for reproducibility
DIOs are needed for attribution

The swh-id : an example for IDOs
Software Heritage archive is using a Merkle tree as data structure

- A combination of a tree and a hash function

This data structure provides a

Questions:
How to maintain the link between the DIO and IDO?

The ascl-id: an example for DIOs
A quick tour of the Astrophysics Source Code Library
Built to improve the transparency, reproducibility, and falsifiability of research
Should be open source so you can look at it
Exposed metadata:

- ASCL ID
- Software name
- AUthors
- Description
- Download site
- Research article
- Bibcode
- Preferred citation
- Keyword
- Number of views

Also have unexposed metadata
Metadata is deliberately kept light
Identifier formula is ascl:yymm.xx
Number of citations is increasing each year
ASCL was first a repository and it become a Repo + Registry
By dropping the requirement to deposit code, though still accepts code deposits

The ASCL pipeline for assigning an identifier:

1. Already in ASCL?
2. Meets ASCL criteria?
3. Conforms to style guide?
4. Editor sequesters entry

Site link curation
Entry curation
Questions:

- We can't have resources like that running on goodwill and coffee
- The problem remains the link to the resources and the sustainability of these resources
    - A lot of research software is written only for the specific time