Archiving, referencing and attributing research software
towards software as a first class citizen

Roberto Di Cosmo
Inria and Université de Paris

February 25th, 2020
Outline

1. Software Source Code: a precious heritage
2. Software Source Code: a (forgotten) pillar of Science
3. Meet Software Heritage
4. Archive and reference all the source code
5. Describe and cite research source code
6. The road ahead
Software source code: a precious part of our heritage

Harold Abelson, Structure and Interpretation of Computer Programs (1st ed.) 1985

“Programs must be written for people to read, and only incidentally for machines to execute.”

Apollo 11 source code (excerpt)

```
P63SP0T3 CA BIT6 # IS THE LR ANTENNA IN POSITION 1 YET
EXTEND RAND CHAN33
EXTEND BZF P63SP0T4 # BRANCH IF ANTENNA ALREADY IN POSITION 1
CAF CODES08 # ASTRONAUT: PLEASE CRANK THE
TC BANKCALL # SILLY THING AROUND
GODPER1
TCF GOTOPO0M # TERMINATE SEE IF HE’S LYING
TCF P63SP0T3 # PROCEED
P63SP0T4 TC BANKCALL # ENTER INITIALIZE LANDING RADAR
CDR SETP0S1
TC POSTJUMP # OFF TO SEE THE WIZARD ...
Cadr BURNBABY
```

Quake III source code (excerpt)

```
float q_sqrt(float number)
{
    long i;
    float x2, y;
    const float threehalves = 1.5F;
    x2 = number * 0.5F;
    y = number;
    i = * ( (long *) &y ); // evil floating point bit level hacking
    i = 0x5F3759df - ( i >> 1 ); // what the fuck?
    y = * ( float * ) &i;
    y = y * ( threehalves - ( x2 * y + y ) ); // 1st iteration
    // y = y * ( threehalves - ( x2 * y * y ) ); // 2nd iteration, this can be removed
    return y;
}
```

Len Shustek, Computer History Museum

“Source code provides a view into the mind of the designer.”
Source code is a special and endangered heritage

Software evolves over time
- projects may last decades
- the development history is key to its understanding

Complexity
- millions of lines of code
- large web of dependencies
  - easy to break, difficult to maintain
- sophisticated developer communities

Precious, endangered Executable and human readable knowledge
key people are passing away …
no organised effort to catalog and archive it
Outline

1. Software Source Code: a precious heritage
2. Software Source Code: a (forgotten) pillar of Science
3. Meet Software Heritage
4. Archive and reference all the source code
5. Describe and cite research source code
6. The road ahead
Software Source code: pillar of Open Science

Software is everywhere in modern research

[…] software […] essential in their fields.

Top 100 papers (Nature, 2014)

Sometimes, if you don’t have the software, you don’t have the data

Christine Borgman, Paris, 2018

Open Science: three pillars

Source code is needed to:

- reproduce and verify,
- modify and evolve, building new experiments from old ones

N.B.: the links in the picture are essential
The state of the art (in CS!) is far from ideal

ICSE (Zannier, Melrik, Maurer, 2006)
- complete absence of replication studies

ACM TOSEM 2001 to 2006
- 60% of all papers have tools: only 20% installable

Collberg’s 2015 study
- 601 mainstream papers: 508 with tools, only 40% installable

Main reasons
- source code (or the right version of it) cannot be found
Where we stand

A wealth of initiatives!

- Policies: ACM Artifact Review and Badging, …
- Working groups: FORCE11, RDA, SPSO, …
- Metrics: Open Science Monitor (Elsevier!), …
- Repositories: FigShare, Zenodo, …

but …

Lack of recognition

not (yet) a first class citizen

- in the EOSC plan
- in the scholarly works

Lack of proper guidance on how to

- *archive* and *reference* software
- choose a license
- *cite* a software project
A plurality of needs

**Researcher**
- archive and reference sw used in articles
- get credit for the software they develop
- verify/reproduce/improve results

**Laboratory/team**
- track software contributions
- produce up-to date report / web page

**University/Research Organization**
- central view of research software assets
- tech transfer
- impact metrics
What is at stake in increasing order of difficulty

**Archival**
Research software artifacts must be properly **archived**
make it sure we can **retrieve** them (**reproducibility**)

**Identification**
Research software artifacts must be properly **referenced**
make it sure we can **identify** them (**reproducibility**)

**Metadata**
Research software artifacts must be properly **described**
make it easy to **discover** them (**visibility**)

**Citation**
Research software artifacts must be properly **cited** (**not the same as referenced!!**)
to give **credit** to authors (**evaluation!!**)

Let’s focus on the **first two!**
1. Software Source Code: a precious heritage
2. Software Source Code: a (forgotten) pillar of Science
3. Meet Software Heritage
4. Archive and reference *all* the source code
5. Describe and cite *research* source code
6. The road ahead
Collect, preserve and share the source code of all the software

Preserving our heritage, enabling better software and better science for all

Reference catalog
find and reference all the source code

Universal archive
preserve all the source code

Research infrastructure
enable analysis of all the source code
An international, non profit initiative built for the long term

Sharing the vision

United Nations Educational, Scientific and Cultural Organization

www.softwareheritage.org/support/testimonials

And many more ...

Donors, members, sponsors

Platinum sponsors

Gold sponsor

Silver sponsors

Bronze sponsors

www.softwareheritage.org/support/testimonials

Roberto Di Cosmo

(CC-BY 4.0) Research Software  February 25th, 2020  9 / 21
Software Heritage

Technology
- transparency and FOSS
- replicas all the way down

Content (billions!)
- intrinsic identifiers
- facts and provenance

Organization
- non-profit
- multi-stakeholder

Roberto Di Cosmo

Largest software archive, principled

Global development history permanently archived in a unique git-like Merkle DAG

- ~400 TB (uncompressed) blobs, ~20 B nodes, ~280 B edges
Outline

1. Software Source Code: a precious heritage
2. Software Source Code: a (forgotten) pillar of Science
3. Meet Software Heritage
4. Archive and reference all the source code
5. Describe and cite research source code
6. The road ahead
Software Heritage: a revolutionary infrastructure

- **universal archive** of all source code
  - we archive *all* software: both research and non research
  - we *proactively collect software* in a systematic way

- **intrinsic identifiers for reproducibility**
  - identify software artefacts *without any third party*
  - cryptographically strong, compatible with git hashes

**Demo**

2012 Parmap paper *before and after*; OCamlP3l paper for the Ten year challenge


See also: Apollo 11 (and the blog post!), Quake III Arena
The SWH-ID schema

```
sw:1:cnt:41ddb23118f92d7218099a5e7a990cf58f1d07fa
```

- `schema_version`
- `object_id`
- `prefix`
- `object_type`
The SWH-ID schema

```
swh:1:cnt:41ddb23118f92d7218099a5e7a990cf58f1d07fa
```

- **prefix**: `swh:1:cnt`
- **object_type**: "cnt" - content
- **object_id**: `41ddb23118f92d7218099a5e7a990cf58f1d07fa`
- **schema_version**
The SWH-ID schema

```
swh:1:cnt:41db23118f92d7218099a5e7a990cf58f1d07fa
```

- **prefix**: `swh`
- **object_type**: `cnt`
- **object_id**: `41db23118f92d7218099a5e7a990cf58f1d07fa`

- **lines_ctx**: `;lines=64-72`
- **origin_ctx**: `;origin=https://github.com/chrislgarry/Apollo-11`
A worked example
A worked example
A worked example

Directories

100644 blob c5baade4c44766042186ef858c0fd63d587ebf09 .gitignore
100644 blob 2d6a34af6f52cf3cf6b0c2f7bd06480bd255e77f AUTHORS
100644 blob 94a9ed824d3859793618152ea559a168bbcb5e2 LICENSE
100644 blob d9b2665a435a43f8a79a84e8867751dfb8957cb MANIFEST.in
100644 blob 524175c2bad0b35b975f79284c2f5a6d5eaf2eb4 Makefile
100644 blob 5c7e3a5b8d8d08682ba7793f440492ed9678bb3 Makefile.local
100644 blob 8617980629c2de4e680404f99aa749685b3e67b README.db_testing
100644 blob 76b29f94fc815e0889c414d38d787ce08ec51c4 README.dev
040000 tree ele10ecef948af0b93adb0372aef89f12e92618a bin
040000 tree 83e56d0beaf77f3c77a45a345c80fcb8af5f30013 debian
040000 tree a34c9c4ba213f0c6ec67f9816348d2795557af5 docs
100644 blob f2a6d32c6135a7827bd76167b01df2ae4f1f53 requirements.txt
100755 blob eee147c36ca0f0b2c2d820da8dc026c5b568180b setup.py
040000 tree 224bb4c1f4c67fca1d160b0f8579e1a4f3 sql
040000 tree 863c19cd77be993168106ab5baf51f40c6300e swh
040000 tree 8fb905b56ba8ed692f1209b27734b7c4c61d6c1 utils

id: 515f00d44e92c65322aaa9bf3fa097c00ddb9c7d
A worked example
Revisions

<table>
<thead>
<tr>
<th>Details</th>
<th>Changes</th>
<th>Files</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHA: 963634dca6ba5dc37e3ee426ba091092c267f9f6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Author: Nicolas Dandrimont <a href="mailto:nicolas@dandrimont.eu">nicolas@dandrimont.eu</a> (Thu Sep 1 14:26:13 2018)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Committer: Nicolas Dandrimont <a href="mailto:nicolas@dandrimont.eu">nicolas@dandrimont.eu</a> (Thu Sep 1 14:26:13 2018)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Subject: provenance.tasks: add the revision -&gt; origin cache task</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parent: fc3a8b59ca1df424d860f2c29ab07fefe4dc35d10</td>
<td>test...storage: properly pipeline origin and cont...</td>
<td></td>
</tr>
<tr>
<td>provenance.tasks: add the revision -&gt; origin cache task</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

```
515f00d44e92c65322aaa9bf3fa097c00ddb9c7d
parent fc3a8b59ca1df424d860f2c29ab07fefe4dc35d10
author Nicolas Dandrimont <nicolas@dandrimont.eu> 1472732773 +0200
committer Nicolas Dandrimont <nicolas@dandrimont.eu> 1472732773 +0200

provenance.tasks: add the revision -> origin cache task

id: 963634dca6ba5dc37e3ee426ba091092c267f9f6
```
A worked example
A worked example

Releases

object c0c9f16b1e1334f593e7567570a1761b156e6eb1d

type commit
tag v0.0.51
tagger Nicolas Dandrimont <nicolas@dandrimont.eu> 1472042163 +0200

Release swh.storage v0.0.51

- Add new metadata column to origin_visit
- Update swh-add-directory script for updated API

---- BEGIN PG SIGNATURE -----

id: 85083a5cc14a441c89dea73f5bdf67c3f9c6afdb

---- END PG SIGNATURE -----

Roberto Di Cosmo

(CC-BY 4.0) Research Software  February 25th, 2020 14 / 21
A worked example
A worked example

Snapshots

commit 08fbee2577010952b6e3c3e21691466c53ad9358 refs/heads/atime
commit ba54443a4e4f0f9e3d3a46c2792bac4f06e16f7eb refs/heads/directory-listing-arrays
commit d90a6b0d18028fe6506b927fbf01ab5d72138d9c5 refs/heads/fo0
commit c7ff9e09a1e67e22f604960f5a8610ef2f7e62ed6e8c0e8 refs/heads/master
commit 7ca197fdce6dd8a24047e54b1ede98b1434631a0f9c2 refs/heads/tmp/directory-listing-arrays
commit 0422e2585f706585a65d427b533e4f0a2525e2e2 refs/heads/tmp/generic-releases
tag 201903c5cb78c7b989665597790f6d0907c72f7555 refs/tags/v0.1.4

tag 72a21991a384e53999eddbb67/fb0b0ee5792ee2cd refs/tags/v0.4.0.10

tag 3599e6c0a6ebdb87e5d3757675af5230b9ff4f7f95c refs/tags/v0.4.0.11

tag 33778a257a4b59ad977870d8d566f2f56556 refs/tags/v0.4.0.12

tag 86f7a4b2753b22c5f903112c2b7f043b8b35d refs/tags/v0.4.0.13

tag 5a63257a6da55c8d8744236df92a1e3273bd refs/tags/v0.4.0.14

tag 58f6b4ad580b0f5f4b051f9367643cb01ac97f refs/tags/v0.4.0.15

tag 8c8b9b85f44998fb5361177429289f60e5e51c refs/tags/v0.4.0.16

tag a542e44e3e89fbed3e5e2871ee36589c9abc7d6 refs/tags/v0.4.0.17

tag 22ba2f1569a1222de55555466e16e6f4993d9 refs/tags/v0.4.0.18

tag 6b0979a4c80d5477f8cd424aadd90c82536e4f7c refs/tags/v0.4.0.19

tag 3d2f5a59f9ca23baade5f1a5f5a03275a67 refs/tags/v0.4.0.20

tag 3147e3d312ec6f649278f30e96b1237abf7c7 refs/tags/v0.4.0.20

tag 215e50da1a11e82ed7e6eb4b6703a87908 refs/tags/v0.4.0.20

tag 3f1b88c2872a5d82521214257a35d6f85ff1a1d refs/tags/v0.4.0.22

tag 8c0b9e80a4d73f5a262789e46b16ac3c72aba4 refs/tags/v0.4.0.23

id: b464cad1b66ff266a37b46ea6e7a04b545e904b
Zoom on the trust model for identifiers

Trust model for usual DOIs

Trust model for DOIs with checksums

Trust model for SWH-IDs
1. Software Source Code: a precious heritage
2. Software Source Code: a (forgotten) pillar of Science
3. Meet Software Heritage
4. Archive and reference all the source code
5. Describe and cite research source code
6. The road ahead
Context

Many articles/guidelines

- reproducibility
- archival
- credit and evaluation

Most common limitations

- software is ’just data’
- citation = reference = DOIs
- citation produced by automated tools

A few remarkable exceptions

- ASCL (since 1999): metadata only, carefully curated
- geodynamics.org: source, documentation, metadata
- swmath.org: software catalog via articles

Software Citation WG at Inria (since 10/2018)

- leverage a 50 year experience, make recommendations
- read more https://hal.archives-ouvertes.fr/hal-02135891
## Why it is not simple

### Software is complex

<table>
<thead>
<tr>
<th>Structure</th>
<th>monolithic/composite; self-contained/external dependencies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lifetime</td>
<td>one-shot/long term</td>
</tr>
<tr>
<td>Community</td>
<td>one man/one team/distributed community</td>
</tr>
<tr>
<td>Authorship</td>
<td>complex set of roles <em>(more later)</em></td>
</tr>
<tr>
<td>Authority</td>
<td>institutions/organizations/communities/single person</td>
</tr>
</tbody>
</table>

### Various granularities

- **Exact status of the source code** for reproducibility, e.g.
  
  "you can find at `swh:1:cnt:cdf19c4487c43c76f3612557d4dc61f9131790a4;lines=146-187` the core algorithm used in this article"

- **(Major) release** "This functionality is available in OCaml version 4"

- **Project** "Inria has created OCaml and Scikit-Learn".
Proposals for the scholarly world

Refined ontology for contributors
- Design, Architecture,
- Coding, Testing, Debugging,
- Documentation, Maintenance, Support,
- Management

see also CRediT, Geodynamics

Reference is distinct from citation
- Reference is for *reproducibility*
- Citation is for *credit*

They must not be conflated

Beware of the numbers game:

… do we really want an *s-index*?

Keep the human in the loop

When *credit* is at stake, automation/crowdsourcing is not enough!

Humans are needed to get *quality information*
First steps with HAL / Software Heritage

How it works, what is special

**Generic mechanism:**
- SWORD based
- review process
- versioning

**Today:** deposit .zip or .tar.gz file (guide)
**Tomorrow:** just provide the SWH id

Deposit/describe research software in HAL

- **author:** [https://hal.archives-ouvertes.fr/hal-01872189](https://hal.archives-ouvertes.fr/hal-01872189)
- **moderator:** [https://hal.archives-ouvertes.fr/hal-01876705](https://hal.archives-ouvertes.fr/hal-01876705)

Examples

LinBox, SLALOM, Givaro, NS2DDV, SumGra, Coq proof, ...
The swmath.org approach

Article based citation

See for example:

- SemiPar on swmath.org
We need to care more about research software

You can help make a change

- leverage Software Heritage in conferences, journals, AEC for archival and reference
  https://www.softwareheritage.org/save-and-reference-research-software/
- join the conversation on software citation and software evaluation criteria
- tackle the scientific problems: big code, classification, infrastructure, etc.

Thank you!

Jean-François Abramatic, Roberto Di Cosmo, Stefano Zacchiroli
Building the Universal Archive of Source Code, CACM, October 2018 (10.1145/3183558)

Roberto Di Cosmo, Morane Gruenpeter, Stefano Zacchiroli
Referencing Source Code Artifacts: a Separate Concern in Software Citation,
CiSE 2020 (10.1109/MCSE.2019.2963148) (hal-02446202)

Pierre Alliez, Roberto Di Cosmo, Benjamin Guedj, Alain Girault, Mohand-Said Hacid, Arnaud Legrand and Nicolas Rougier
Attributing and referencing (research) software: Best practices and outlook from Inria,
CiSE 2020 (10.1109/MCSE.2019.2949413) (hal-02135891)
Appendix
Reference platform for *Big Code*

- unique *observatory* of all software development
- big data, machine learning paradise: classification, trends, coding patterns, code completion…

First datasets are available!

- full graph of software development (~20Bn nodes, ~200Bn edges) see Pietri, Spinellis, Zacchirolı, MSR 2019
  
  [https://dx.doi.org/10.1109/MSR.2019.00030](https://dx.doi.org/10.1109/MSR.2019.00030)

Raising awareness about Software Source Code

UNESCO, Inria, Software Heritage invite 40 international experts meet in Paris …

Their call is published on Feb 2019

It’s an important policy tool, already referenced and used …  yes, you can sign it!

7 Guidelines detailed and SWH-ID use cases

8 News

9 Inria’s commitment

10 Identifiers are not easy
Prepare your software source code

Prepare your public repository with:
- README, LICENSE, AUTHORS & codemeta.json files

What’s a good README

**extracted from Eric Steven Raymond and Make a README**

*MUST* include:
- **Name** and a **description** of the software.

*SHOULD* include:
- how to **run** and **use** the source code
- build **environment**, installation, requirements

*CAN* include:
- project **website** or **documentation** pointer and recent **news**
- **visuals**
Submit save request on SWH

Save code now on https://archive.softwareheritage.org/save/

- git, svn or mercurial
- intrinsic metadata files
- complete history
Reference software artifacts in your articles

Choose the granularity level for the reference:

**file (with code fragment)**

```markdown
swh:1:cnt:c60366bc03936e6e6e6509b23307321f1035e23;lines=473-537
... and add ;origin=https://github.com/sagemath/sage/
```

James McCaffrey’s *algorithm* in sagemath

**directory**

```markdown
swh:1:dir:c6f07c2173a458d098de45d4c459a8f1916d900f
... and add ;origin=https://github.com/id-Software/Quake-III-Arena/
```

source code of *Quake-III Arena* from id-Software
Reference software artifacts in your articles

**specific release**

```
swh:1:rel:22ece559cc7cc2364edc5e5593d63ae8bd229f9f
... and add ;origin=https://github.com/darktable-org/darktable/
```

**release 2.3.0** of Darktable, dated 24 December 2016

**full snapshot (including all branches and all releases)**

```
swh:1:snp:c7c108084bc0bf3d81436bf980b46e98bd338453
... and add ;origin=https://github.com/darktable-org/darktable/
```

**a snapshot** of the entire Darktable repository (4 May 2017, GitHub)
News: archiving public code

https://code.etalab.gouv.fr
Paris Call on Software Source Code

“We call to support efforts to gather and preserve the artifacts and narratives of the history of computing, while the earlier creators are still alive”

SWHAP: an important step forward

- detailed guidelines to curate landmark legacy source code and archive it on Software Heritage
- intense cooperation with Università di Pisa and UNESCO
- open to all, we’ll promote it worldwide

https://www.softwareheritage.org/swhap
Thomas Jefferson, February 18, 1791

…let us save what remains: not by vaults and locks which fence them from the public eye and use in consigning them to the waste of time, but by such a multiplication of copies, as shall place them beyond the reach of accident.

Welcoming ENEA

- first institutional mirror
- increased resilience
- AI infrastructure for researchers
- stepping stone to an European joint effort
7 Guidelines detailed and SWH-ID use cases
8 News
9 Inria’s commitment
10 Identifiers are not easy
Inria’s ongoing contributions

Software Heritage
universal archive (research) software source code archived and referenced

Reproducibility
tools Guix (now with Software Heritage)
training/research RR workshops, MOOC

Research software curation
HAL - SWH bridge curation of metadata, and deposit in Software Heritage
Outline

7 Guidelines detailed and SWH-ID use cases

8 News

9 Inria’s commitment

10 Identifiers are not easy
URL decay disrupts the web of reference

Web links are not permanent (even permalinks)

*there is no general guarantee that a URL… which at one time points to a given object continues to do so*

*T. Berners-Lee et al. Uniform Resource Locators. RFC 1738.*

URLs used in articles decay!

Analysis of *IEEE Computer* (Computer), and the *Communications of the ACM* (CACM): 1995–1999

- the *half-life* of a referenced URL is *approximately 4 years* from its publication date


How Do Astronomers Share Data?

Pepe, Goodman, Muench, Crosas, Erdmann
dx.doi.org/10.1371/journal.pone.0104798

---

PLOS August 28, 2014
DOI limitations

Example: doi:10.1109/MSR.2015.10

- to find what 10.1109/MSR.2015.10 is, go to a resolver (e.g. doi.org)
- this returns http://ieeexplore.ieee.org/document/7180064/
- at this URL we find …

Architecture of the DOI infrastructure

- DOI resolution can change
- content at URL can change
- no intrinsic way of noticing
- persistence based on good will of multiple parties