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

Roberto Di Cosmo

September 24th, 2019

Software Heritage
THE GREAT LIBRARY OF SOURCE CODE
Short Bio: Roberto Di Cosmo

Computer Science professor in Paris, now working at INRIA

- 30 years of research (Theor. CS, Programming, Software Engineering, Erdos #: 3)
- 20 years of Free and Open Source Software
- 10 years building and directing structures for the common good

1999 DemoLinux – first live GNU/Linux distro
2007 Free Software Thematic Group
    150 members 40 projects 200Me
2015 Software Heritage at INRIA
2018 National Committee for Open Science, France
1 Introduction

2 Software Source Code: a (forgotten) pillar of Science

3 Software Heritage for archival and reference

4 Towards Software Citation

5 The road ahead
Source code is *special*

**Executable and human readable knowledge**

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

Harold Abelson

**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*

---

Matplotlib library
Python dependencies
Real dependencies
Fake OS dependencies induced by package granularity

Roberto Di Cosmo (CC-BY 4.0) Research Software September 24th, 2019 2 / 18
So/f_tware 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

Nota bene

The links in the picture are essential
The state of the art is not ideal

Analysis of 613 papers

- 8 ACM conferences: ASPLOS’12, CCS’12, OOPSLA’12, OSDI’12, PLDI’12, SIGMOD’12, SOSP’11, VLDB’12
- 5 journals: TACO’9, TISSEC’15, TOCS’30, TODS’37, TOPLAS’34

all very practical oriented

The basic question

can we get the code to build and run?
... that’s a whopping 40% of non reproducible works!

The main reasons

source code (or the right version of it) cannot be found
Pressure to make the source code available is raising

Why

Necessary to

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

When and where

- debate started end of first 2000 decade (biology, statistics, medicine, etc.)
- growing in Computer Science since the *ESEC/FSE 2011 Artifact Evaluation context* (winner: Vouillon and Di Cosmo)
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

Roberto Di Cosmo (CC-BY 4.0) Research Software September 24th, 2019 7 / 18
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 **referred**
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!**)
Outline

1. Introduction
2. Software Source Code: a (forgotten) pillar of Science
3. Software Heritage for archival and reference
4. Towards Software Citation
5. 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
The largest software source code archive ever

20 billions intrinsic identifiers for reproducibility
See DIO vs IDO in bit.ly/swhpipdpaper

Reference archive
See the work done at swmath.org

SWH IDs now a standard for Wikidata
See https://www.wikidata.org/wiki/Property:P6138

Policy
Now part of the French National Plan for Open Science
So/ftware Heritage for Open Science

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
- also save code now and curated deposit (e.g. via HAL)

Guidelines are now available

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, Spinelli, Zacchirolì, MSR 2019
  [https://dx.doi.org/10.1109/MSR.2019.00030](https://dx.doi.org/10.1109/MSR.2019.00030)
Demo time

Exploring the archive

- Apollo 11
- Quake III Arena

Feature highlights

- Save code now

Archive interconnect

- SemiPar on swmath.org
Outline

1. Introduction
2. Software Source Code: a (forgotten) pillar of Science
3. Software Heritage for archival and reference
4. Towards Software Citation
5. 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

- **Structure**: monolithic/composite; self-contained/external dependencies
- **Lifetime**: one-shot/long term
- **Community**: one man/one team/distributed community
- **Authorship**: complex set of roles (*more later*)
- **Authority**: institutions/organizations-communities/single person

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*
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
1 Introduction

2 Software Source Code: a (forgotten) pillar of Science

3 Software Heritage for archival and reference

4 Towards Software Citation

5 The road ahead
Conclusion

Research software
- pillar of open science
- finally in the limelight

Doing it right is not easy
- simplistic approaches, "just data", ...
- soon part of research evaluation

You can help make a change
- leverage Software Heritage in conferences and journals for archival and reference
- join the conversation on software citation and software evaluation criteria

Thank you!

Jean-François Abramatic, Roberto Di Cosmo, Stefano Zacchiroli
Building the Universal Archive of Source Code
Communication of the ACM, October 2018

Roberto Di Cosmo, Morane Gruenpeter, Stefano Zacchiroli
Identifiers for Digital Objects: the Case of Software Source Code Preservation
iPRES 2018: Intl. Conf. on Digital Preservation
Appendix
full development history permanently archived!
Outline

6 Under the hood

7 Identifiers are not easy

8 Looking for the right identifiers
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


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


Scholar roster of broken links

An example from Astronomy

<table>
<thead>
<tr>
<th>Domain</th>
<th>links (broken)</th>
<th>html</th>
<th>txt</th>
<th>dat</th>
<th>.gz</th>
<th>tar</th>
<th>jrn</th>
<th>bibtex</th>
</tr>
</thead>
<tbody>
<tr>
<td>techreport.edsu</td>
<td>502 (100)</td>
<td>316 (63)</td>
<td>0</td>
<td>0</td>
<td>4 (2)</td>
<td>5 (0)</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>berkeley.gsfc.nasa.gov</td>
<td>640 (10)</td>
<td>415 (27)</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><a href="http://www.stsci.edu">www.stsci.edu</a></td>
<td>259 (61)</td>
<td>265 (29)</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>15 (16)</td>
</tr>
<tr>
<td>scihub.edsu</td>
<td>472 (152)</td>
<td>212 (99)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>11 (1)</td>
</tr>
<tr>
<td>scireport.sns.gov</td>
<td>457 (194)</td>
<td>125 (74)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><a href="http://www.ics.uci.edu">www.ics.uci.edu</a></td>
<td>362 (64)</td>
<td>277 (31)</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>54 (17)</td>
</tr>
<tr>
<td>gchive.stsci.edu</td>
<td>320 (10)</td>
<td>317 (15)</td>
<td>2</td>
<td>1 (1)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><a href="http://www.isccp.ucar.edu">www.isccp.ucar.edu</a></td>
<td>285 (14)</td>
<td>265 (13)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><a href="http://www.astro.uva.edu">www.astro.uva.edu</a></td>
<td>221 (13)</td>
<td>13 (7)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>7 (5)</td>
</tr>
<tr>
<td>space.mit.edu</td>
<td>193 (10)</td>
<td>56 (5)</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2 (1)</td>
</tr>
<tr>
<td><a href="http://www.astrophysics.edu">www.astrophysics.edu</a></td>
<td>186 (10)</td>
<td>163 (7)</td>
<td>1</td>
<td>16</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td><a href="http://www.nasa.gov">www.nasa.gov</a></td>
<td>188 (10)</td>
<td>94 (12)</td>
<td>1 (1)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4 (1)</td>
</tr>
<tr>
<td><a href="http://www.ipac.caltech.edu">www.ipac.caltech.edu</a></td>
<td>163 (1)</td>
<td>38 (1)</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><a href="http://www.cds.org">www.cds.org</a></td>
<td>196 (1)</td>
<td>166 (1)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><a href="http://ned.ipac.caltech.edu">http://ned.ipac.caltech.edu</a></td>
<td>125 (17)</td>
<td>42 (17)</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>20 (16)</td>
</tr>
<tr>
<td>physics.rit.edu</td>
<td>125 (1)</td>
<td>63 (2)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><a href="http://www.loss.rit.edu">www.loss.rit.edu</a></td>
<td>120 (1)</td>
<td>50 (2)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><a href="http://www.sunspot.astro">www.sunspot.astro</a></td>
<td>118 (11)</td>
<td>22 (10)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1 (1)</td>
</tr>
<tr>
<td><a href="http://www.astro.phys.nus.edu">www.astro.phys.nus.edu</a></td>
<td>115 (11)</td>
<td>43 (16)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>53 (13)</td>
</tr>
<tr>
<td>astroframe.orf</td>
<td>113 (12)</td>
<td>98 (22)</td>
<td>3 (3)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1 (1)</td>
</tr>
</tbody>
</table>

This table lists total number of links and broken links (FTP status codes 2xx, 4xx, and 5xx) to top domain (domains with over 100 links) found within articles published in the four main astronomy journals between 1997 and 2008. The table also shows, for each domain, the portion of links to common filename extensions, as well as links that contain the title character.

The table also shows, for each domain, the portion of links to common filename extensions, as well as links that contain the title character.

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
A system of identifiers is

- a set of labels (the identifiers)
- mechanisms to perform:

<table>
<thead>
<tr>
<th>Generation (minting)</th>
<th>create a new label</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assignment</td>
<td>associate label to object</td>
</tr>
<tr>
<td>Retrieval</td>
<td>get object from a label</td>
</tr>
</tbody>
</table>

- optionally, mechanisms to perform:

<table>
<thead>
<tr>
<th>Verification</th>
<th>check label and object</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reverse Lookup</td>
<td>get label from an object</td>
</tr>
<tr>
<td>Description</td>
<td>get metadata of an object</td>
</tr>
</tbody>
</table>
Mechanisms offered in some systems of identifiers

<table>
<thead>
<tr>
<th>Mech. / System</th>
<th>Handle</th>
<th>DOI</th>
<th>Ark</th>
<th>PURL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generation</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Assignment</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Retrieval</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Verification</td>
<td>N.A.</td>
<td>N.A.</td>
<td>N.A.</td>
<td>N.A.</td>
</tr>
<tr>
<td>Reverse Lookup</td>
<td>N.A.</td>
<td>N.A.</td>
<td>N.A.</td>
<td>N.A.</td>
</tr>
<tr>
<td>Description</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>N.A.</td>
</tr>
</tbody>
</table>
Our challenges in the PID landscape

Typical properties of systems of identifiers
- uniqueness, non ambiguity, persistence, abstraction (opacity)

Key needed properties from our use cases
- gratis: identifiers are free (billions of objects)
- integrity: the associated object cannot be changed (sw dev, reproducibility)
- no middle man: no central authority is needed (sw dev, reproducibility)

we could not find systems with both integrity and no middle man!
An important distinction: DIOs vs. IDOs

The term “Digital Object Identifier” is construed as “digital identifier of an object,” rather than “identifier of a digital object.”

**DIO (Digital Identifier of an Object)**
- digital identifiers for (potentially) non digital objects
  - epistemic complexity (manifestations, versions, locations, etc.)
  - need an authority to ensure persistence and uniqueness

**IDO (Identifier of a Digital Object)**
- digital identifiers (only) for digital objects
  - can provide both integrity and no middle man
  - broadly used in modern software development (git, etc.)

For the core Software Heritage archive, IDOs are enough.
Merkle tree (R. C. Merkle, Crypto 1979)

Combination of
- tree
- hash function

Classical cryptographic construction

fast, parallel signature of large data structures, built-in deduplication
- satisfies all three criteria: gratis, integrity, no middle man!
- widely used in industry (e.g., Git, nix, blockchains, IPFS, …)
IDOs in Software Heritage: a worked example
Sha1: 8624bcdae55baeef...
Sha256: 8ceb4b9ee5aded...
Sha1_git: 94a9ed024d385...
Length: 35147
IDs in Software Heritage: a worked example
Directories

```
100644 blob c5baade4c44766042186ef858c0fd63d587ebf09 .gitignore
100644 blob 2d6a34af6f52cfc3cf6b0c27bd0648fb25e7f7 AUTHORS
100644 blob 94a9ed824d3859793618152ea559a168bbcb85e2 LICENSE
100644 blob d9b2665a435a4f8a79a84e867751dfb9957cbf MANIFEST.in
100644 blob 524175c2bad0635b975f79284c2f5a6d5eaf2eb4 Makefile
100644 blob 5c7e3a5bbdd0b38682ba7793f440492ed9678bb Makefile.local
100644 blob 8617980629c24e680404f99a7496885b3e67b README.db_testing
100644 blob 76b29f94cf815e0869c414d387d87ce08ec514e README.dev
040000 tree e1e10ecef948af0b93adb0372aafc89f12e92618a bin
040000 tree 8e56d8beaf7793c77a45a435c80fcb8af503013 debian
040000 tree a34c9c4ba213f0ced67f9816348d2795557af5 docs
100644 blob f2a6d326c135a7a2787bd76167b01df2ae4f1539 requirements.txt
100755 blob eee147c36caflbcbc2d820da8dc026cb568180c setup.py
040000 tree 224bb4cf4c67fca1d160b6ff2d26094e7e1abf3 sql
040000 tree 8631c9cd77bb993168170ab5ba51f40c6300be swh
040000 tree 8fb905b56ba8ed692f1209b2773b474c61d66c1 utils
```

id: 515f00d44e92c65322aa9bf3fa097c00ddb9c7d
IDOs in Software Heritage: 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: fc3a8b59ca1df424d860f2c29ab07fee4dc35d10</td>
<td>test...storage: properly pipeline origin and cont...</td>
<td></td>
</tr>
<tr>
<td></td>
<td>provenance.tasks: add the revision -&gt; origin cache task</td>
<td></td>
</tr>
</tbody>
</table>

```
515f00d44e92c65322aaa9bf3fa097c00dddb9c7d
parent fc3a8b59ca1df424d860f2c29ab07fee4dc35d10
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
```
Releases

- Add new metadata column to origin.visit
- Update swh-add-directory script for updated API

---BEGIN POP SIGNATURE-----
OBJECT cbc9f16b1e134f593e7567570a1761b156e6eb1d
TYPE commit
TAG v0.0.51
tagger Nicolas Dandrimpont <nicolas@dandrimpont.eu> 1472042163 +0200
Release swh.storage v0.0.51

---END POP SIGNATURE-----

id: 85083a5cc14a441c89ded73f5bf67c3f9c6afdb
IDOs in Software Heritage: a worked example
Snapshots

commit 00fbeb2577010952eb3c21691466c53a1d9158 refs/heads/atime
commit ba5443a324e3f9e323a46c6292ce4f061c67eb refs/heads/directory-listing-arrays
commit d6e09b4d283f3e569b97fe1b1c05e27238d9c5 refs/heads/fo
commit c77f9e9aebb2b27b946907f5a5860f67de468ec8 refs/heads/master
commit 7eca197f1b46d2b284a47e5b4b1ed98e844361a8f2c refs/heads/tmp/addr
commit 642a285f515e8505a650427b553ee4f0252e82e refs/heads/tmp/generated

Id: b464cad1b66ff266a37b46ea6e7a04b545e904b

Roberto Di Cosmo

(CC-BY 4.0) Research Software  September 24th, 2019 11 / 12

<table>
<thead>
<tr>
<th>ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>swh:1:cnt:94a9ed024d3859793618152ea559a168bbcbbb5e2</td>
<td>full text of the GPL3 license</td>
</tr>
<tr>
<td>swh:1:dir:d198bc9d7a6bcf6db04f476d29314f157507d505</td>
<td>Darktable source code</td>
</tr>
<tr>
<td>swh:1:rev:309cf2674ee7a0749978cf8265ab91a60aea0f7d</td>
<td>a revision in the development history of Darktable</td>
</tr>
<tr>
<td>swh:1:rel:22ece559cc7cc2364edc5e5593d63ae8bd229f9f</td>
<td>release 2.3.0 of Darktable, dated 24 December 2016</td>
</tr>
<tr>
<td>swh:1:snp:c7c108084bc0bf3d81436bf980b46e98bd338453</td>
<td>a snapshot of the entire Darktable repository (4 May 2017, GitHub)</td>
</tr>
</tbody>
</table>

Current resolvers: [archive.softwareheritage.org](http://archive.softwareheritage.org) and [n2t.org](http://n2t.org)