Archive and Reference Source Code with Software Heritage

a stepping stone for reproducibility

Roberto Di Cosmo
Director, Software Heritage
Inria and Université Paris Cité
Software **Source Code** is Precious Knowledge

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.”

<table>
<thead>
<tr>
<th>Apollo 11 source code (excerpt)</th>
<th>Quake III source code (excerpt)</th>
</tr>
</thead>
<tbody>
<tr>
<td>P63SP0T3 CA BIT6 # IS THE LR ANTENNA IN POSITION 1 YET</td>
<td><code>float Q_rsqr( float number )</code></td>
</tr>
<tr>
<td>EXTEND RAND CHAN33 EXTEND</td>
<td><code>{</code></td>
</tr>
<tr>
<td>BZF P63SP0T4 # BRANCH IF ANTENNA ALREADY IN POSITION 1</td>
<td><code>long i;</code></td>
</tr>
<tr>
<td>CAF CODE5000 # ASTRONAUT: PLEASE CRANK THE</td>
<td><code>float x2, y;</code></td>
</tr>
<tr>
<td>TC BANKCALL # SILLY THING AROUND</td>
<td><code>const float threehalfs = 1.5F;</code></td>
</tr>
<tr>
<td>CADR GOPERF1</td>
<td><code>x2 = number * 0.5F;</code></td>
</tr>
<tr>
<td>TCF GOTOPOOH # TERMINATE</td>
<td><code>y = number;</code></td>
</tr>
<tr>
<td>TCF P63SP0T3 # PROCEED SEE IF HE'S LYING</td>
<td><code>i = * ( long * ) &amp;y; // evil floating point bit level hacking</code></td>
</tr>
<tr>
<td>P63SP0T4 TC BANKCALL</td>
<td><code>i = 0x5f3759df - ( i &gt;&gt; 1 ); // what the fuck?</code></td>
</tr>
<tr>
<td>CADR SETPO51 # ENTER INITIALIZE LANDING RADAR</td>
<td><code>y = * ( float * ) &amp;i;</code></td>
</tr>
<tr>
<td>TC POSTJUMP # OFF TO SEE THE WIZARD...</td>
<td><code>y = y * ( threehalfs - ( x2 * y * y ) ); // 1st iteration</code></td>
</tr>
<tr>
<td>CADR BURNBABY</td>
<td><code>// y = y * ( threehalfs - ( x2 * y * y ) ); // 2nd iteration, this can be removed</code></td>
</tr>
</tbody>
</table>

Len Shustek, *Computer History Museum* 2006

“Source code provides a view into the mind of the designer.”
A lightning fast growth

<table>
<thead>
<tr>
<th>Apollo 11 (~60,000 lines), 1969</th>
<th>Linux Kernel: 20+ million lines...</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;When I first got into it, nobody knew what it was that we were doing. It was like the Wild West.&quot;</td>
<td>![Graph showing growth of lines of source code over time]</td>
</tr>
<tr>
<td>Margaret Hamilton</td>
<td>... now in your pockets!</td>
</tr>
</tbody>
</table>

Open source software is eating the software world

tens of millions of developers collaborate on open source software worldwide today

Reuse is the new rule

80% to 90% of a new application is... just reuse! (Sonatype survey, 2017)
Source code is \textit{special}: software is \textit{not} data

**Software evolves over time**

- projects may last decades
- the \textit{development history} is key to its \textit{understanding}

**Complexity**

- \textit{millions} of lines of code
- large web of dependencies
  - easy to break, difficult to maintain
  - \textit{research software} a thin top layer
- sophisticated \textit{developer communities}

**The human side**

design, algorithm, code, test, documentation, community, funding, and so much more...
Software is a pillar of Open Science

**Software powers all research disciplines!**
Proportion of French publications mentioning use of code or software, by discipline (2024 data from https://barometredelascienceouverte.esr.gouv.fr/)

<table>
<thead>
<tr>
<th>Discipline</th>
<th>Proportion of Publications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Biologie fondamentale</td>
<td>34%</td>
</tr>
<tr>
<td>Informatique et systèmes de l'information</td>
<td>41%</td>
</tr>
<tr>
<td>Sciences de la Terre, mécanique, énergie</td>
<td>45%</td>
</tr>
<tr>
<td>Ingénieurs</td>
<td>47%</td>
</tr>
<tr>
<td>Mathématiques</td>
<td>21%</td>
</tr>
<tr>
<td>Sciences sociales</td>
<td>19%</td>
</tr>
<tr>
<td>Sciences humaines</td>
<td>18%</td>
</tr>
</tbody>
</table>

**A key pillar of Open Science: software (source code)**

→ Access to the source code is essential!
→ Preserving (the history of) source code is necessary for reproducibility

**How are we handling software and source code in research?**
Reproducibility in Computer Science: state of the art in the field ~2010

Software Engineering

2009: Carlo Ghezzi, 60% of ACM TOSEM papers have code, only 20% installable

Computer systems research

2014: Christian Collberg, analysis of ~600 papers in prestigious venues, ~200 cannot even find the source code!
Awareness and actions

Artifact Evaluation Committees

2011: run the first time as an award at ESEC-FSE (J. Vouillon and R. Di Cosmo)

2012-today: the process generalizes to a list too long to maintain

ACM software badges for publications

See home page for details.

- Very good intentions, but …
- Perfectible implementation
A few key issues in reproducibility *(there are many more!)*

**Archive**

Ensure *long term availability* of artifacts *with the development history*

**Reference**

Ensure *precise identification* of artifacts at *various levels of granularity*

**Describe**

Provide *detailed description* *(machine readable metadata)*

and *proper documentation* *(build instructions, dependencies, configuration)*

and also *link to relevant papers*
Not there yet, event for these most basic needs - ACM DL

Only a DOI identifier …

… does not fill the author’s needs

Zip file with source code, losess the version control history!
Not there yet, event for these basic needs: Papers with code

The Forward-Forward Algorithm: Some Preliminary Investigations

nebuly-ai/nebullvm • PyTorch • NA 2022

The aim of this paper is to introduce a new learning procedure for neural networks and to demonstrate that it works well enough on a few small problems to be worth further investigation.
Forges are not archives!

Google begins shutdown of its code repository

After nine years, Google's open-source code repository, Google Code, started closing shop today by disabling new projects and announcing the permanent shut down of the platform.

deleGoogle code;

Sunsetting Mercurial support in Bitbucket

April 21, 2020 | 3 min read

[Update Aug 26, 2020] All hg repos have now been disabled and cannot be accessed.

[Update July 1, 2020] Today, mercurial repositories, snippets, and wikis will turn to read-only mode. After July 8th, 2020 they will no longer be accessible.

Code collaboration platform GitLab acquires rival Gitorious, will shut it down on June 1

Over 1 million projects gone?
We need a universal archive
Meet Software Heritage!

Ensures availability guarantees integrity enables traceability

500+ platforms
All versions history in a single graph

- $35 \times 10^9$ nodes
- $500 \times 10^9$ edges
~ $1.5$ PB storage
We can and must do better: **archive** in Software Heritage

- Regular crawling
- **One click** archival via updateswh browser extension
- Webhooks for BitBucket, Gitea, GitHub, GitLab, Sourceforge
We can and must do better: **reference** in Software Heritage

**SWHID**
Cryptographic, intrinsic

**SoftWare Hash IDentifier**

All levels of **granularity**  
Rich **contextual** (optional) information

Integrity of identified artifact can be verified independently

Standard at [https://swhid.org](https://swhid.org)
We can and must do better: **reference** in Software Heritage

Getting the SWHID for a code fragment

You can also get the SWHID of a file, or a code fragment inside a file. For this, navigate first to the file, select (optionally) the code fragment of interest by clicking on the line number of the first line, and shift-clicking on the line number of the last line. Then, pull out the red Permalinks tab and copy the SWHID identifier or the corresponding permalink.

~ 30 billion SWHIDs can be found in Software Heritage

All levels of **granularity**:
- repository snapshot
- release
- revision
- directory
- file content
- code fragment
We can do so much better: **reference** in Software Heritage

HOWTO with animations:


**Step 1: prepare your public repository**

- add a README file
- add an AUTHORS file
- add license information in one of the two recommended ways
  - a LICENSE file at the root of your project, or
  - a LICENSES directory containing all the licenses used in your project, and an SPDX compliant copyright header in all your source code files (see the REUSE instructions for details and tools)
- (optionally) add a codemeta.json file containing machine readable metadata (can be produced using the CodeMeta Generator)
A few adoption indicators

Policy

- Recommendations in ANR 2023 guidelines (p. 17)
- HAL+SWH in the Open Science software booklet

Users and collaborations

What are they “referencing”?

<table>
<thead>
<tr>
<th>source</th>
<th>n</th>
<th>percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not available</td>
<td>2868</td>
<td>46.72</td>
</tr>
<tr>
<td>GitHub</td>
<td>1151</td>
<td>18.55</td>
</tr>
<tr>
<td>software heritage</td>
<td>387</td>
<td>6.24</td>
</tr>
<tr>
<td>zendo</td>
<td>142</td>
<td>2.29</td>
</tr>
<tr>
<td>r package</td>
<td>70</td>
<td>1.13</td>
</tr>
<tr>
<td>cran</td>
<td>56</td>
<td>0.90</td>
</tr>
<tr>
<td>r package version</td>
<td>54</td>
<td>0.87</td>
</tr>
<tr>
<td>gitlab</td>
<td>35</td>
<td>0.56</td>
</tr>
</tbody>
</table>

Projects

- EOSC FAIRCORE4EOSC
- EOSC FAIR-IMPACT
- The CodeMeta Project

Graphics Replicability Stamp Initiative

b/Surf: Interactive Bézier Splines on Surface Meshes
Claudio Mencini, Giacomo Nazzaro, Fabio Pelosini, Enrico Popo
IEEE Transactions on Visualization and Computer Graphics (TVCG)
In France: HAL + Software Heritage for describe and cite

https://hal.archives-ouvertes.fr/ha

swh:1:dir:393b611a1424f032e83569bf6762371cfcf65
Demo time

- **Browse and Reference** (e.g. Apollo 11 [excerpt], your work may be already there !)

- **Trigger archival**, use the updateswh browser extension, configure the webhooks

- **Cite with biblatex-software** (CTAN, Overleaf ACMART template)

- **Describe with Codemeta** (use codemeta generator)

- **Curated deposit in SWH via HAL**, see for example: LinBox, SLALOM, Givaro, NS2DDV, SumGra, Coq proof, …

- **Extracting all the software products** for Inria, for CNRS, for CNES, for LIRMM or for Rémi Gribonval using HalTools

- **Example with Parmap**: devel on Github, archive in SWH, curated deposit in HAL

- **Example research articles**:
  - compare Fig. 1 and conclusions in the 2012 version and the updated version
  - SWHID in a replication experiment
Latest news: SOFair and SCOSS

Software Heritage is an open non-profit infrastructure for archiving, referencing and sharing software source code, launched by Inria in 2016, in partnership with UNESCO.

Archiving over 260 million software projects already, it is built according to the UNESCO recommendations for Open Science: open, multi-stakeholder, non-profit, using exclusively open-source components, it serves as a cornerstone for Open Science.

It simplifies the deposit of research software and associated metadata, amplifying the visibility and impact of scholarly outputs. Researchers take advantage of Software Heritage's vast collection of software projects, that enables citation, referencing and sharing of software artifacts, improving reproducibility and traceability of research. Libraries benefit from Software Heritage's robust infrastructure, which offers long-term archival and unique identification of software, removing the need for custom and in-house archival solutions.

By supporting Software Heritage, you're supporting unfettered access, reference and citation of software produced by academic research, reinforcing the principles of open science.

WHY HAS IT BEEN DEEMED AN ESSENTIAL INFRASTRUCTURE?

The SCOSS Board considers Software Heritage to be an essential open science infrastructure because it provides continued access to the software and code outputs produced by researchers globally.

SCOSS FUNDING TARGET

€ 900,000
The way ahead

Archival and reference for source code

- **Technical barriers** are mostly solved issues (*over 6 years of work*)
- **Social barriers** still stand in the way (adoption, training, cost mutualization, …)

Thank you

- Software Heritage: [https://softwareheritage.org](https://softwareheritage.org) and [the 2022 annual report](https://softwareheritage.org/annual-report-2022)
- Software deposit and metadata curation: [HAL-SWH Webinar, July 2022](https://hal.archives-ouvertes.fr/hal-01826380)
- Deuxième plan national pour la Science Ouverte: [official website](https://www.societesscientifiques.org/)
- Software Pillar session in OSEC 2022: [official website](https://www.osec2022.org/)

Learn more

- Roberto Di Cosmo and Marco Danelutto. [Rp] Reproducing and replicating the OCamlP3I experiment. ReScience C, 6(1):#2, April 2020. [link]