Software Heritage, the universal source code archive

Roberto Di Cosmo,
Software Heritage, INRIA and Paris University

Abstract:
All activities in modern societies hinge on software. The source codes of software programs contain a growing share of our technical, scientific and organizational knowledge. As a part of our cultural heritage, they have to be preserved. Software Heritage has taken up this challenge. It is building a universal archive for storing software source codes, a common good to be made available to everyone. The task is vast; the stakes are immense. For one thing, the past of software must be preserved, and this requires considerable research efforts. For another, a major tool has to be built for observing current software development and improving future programs. The intent is to construct an international infrastructure to preserve this common good while respecting national sovereignty.

Within a few decades, software has become the driving force in industry, the gasoline of innovation, the key to communications, transactions or operations of any type, to organizing our society and forming our political opinions. Software programs control systems on board our means of transportation or used for communications, or commercial and financial transactions. It is the core of medical equipment and devices. It sees to the proper operation of transportation and communication networks, the banking system and financial establishments. Software, whether for mobile devices or the cloud, is crucial to the operation of economic, social and political organizations, whether public or private. Furthermore, it is indispensable for accessing electronic information. Along with articles and data, it is one of the pillars of modern research (NOORDEN *et al*. 2014). As a consequence, software is a major part of our scientific, technical industrial heritage. All this raises major strategic questions.[1]

Upon a closer look, we soon realize that the actual knowledge contained in software programs is found not in the executable files but in the "source code". According to the definition in the General Public License (GPL),"*The source code for a work means the preferred form of the work for making modifications to it*" (GNU 2017). The source code, this special form of knowledge, is made to be UNDERSTOOD by human beings (developers), but it can be mechanically translated into a form directly EXECUTABLE on a machine. The very terms used by the computer community are telling: "programming languages" are used to "write" software: "*programs must be written for people to read, and only incidentally for machines to execute*" (ABELSON *et al*. 1985). Source code is clearly a human creation like other written documents; and software developers deserve as much respect as authors.

[1] This article, has been translated from French by Noal Mellott (Omaha Beach, France). The translation into English has, with the editor's approval, completed a few bibliographical references. All websites were consulted in December 2020.

A final point: the source code of software programs is increasingly complex. Groups of developers work together to modify it and regularly issue updates. To understand the source code, the history of the development efforts put into making software has to be accessible.

The source code of software programs is, therefore, a highly valuable heritage, as stated by Len Shustek in his excellent article in 2006 and by Donald Knuth in 1984. It must be preserved. This is the mission of Software Heritage, launched in 2015 with INRIA's backing:[2] to collect, organize, preserve and make it easy to access all source codes publicly on the planet, regardless of where and how the code was developed and distributed. The goal is to build a common infrastructure for the long-term preservation of source code from destruction. In addition, this infrastructure will be available for large-scale studies of the code and for current development projects. It will thus be turned toward a better future.

A complicated task

Archiving all available source codes is a complicated assignment. As has been pointed out (ABRAMATIC *et al*. 2018), different strategies have to be pursued depending on whether we are trying to collect open or proprietary source code. Furthermore, the source code readily available on line is not to be handled in the same way as source code on an old physical medium. For the latter, a process of "computer archeology" has to be used, as INRIA has started doing in a program in cooperation with Pisa University and UNESCO. This program has produced SWHAP, which is used to find, document and archive outstanding software in the history of computer science in Italy.[3] For open source code easy to obtain on line, the most appropriate approach is to build a "harvesting machine" that automatically collects contents from a variety of repositories (*e.g.*, GitHub, GitLab.com or BitBucket) and platforms that distribute package software (*e.g.* Debian, CRAN, Pypi or NPM).

Although this might, at first sight, seem similar to the approach used for Web archives, we soon see, upon a closer look, that the task is much more difficult. First of all, since there is no standard protocol, "adaptors" (rightly called "listers") have to be built for each platform that develops and distributes software in order to extract the list of software projects hosted there. We then come face-to-face with a brood of data formats and models that have been used during the history of the software's development, whence a major problem: how to be sure that this history will be "legible" in the future, even once the revision-control tools (*e.g.*, Git, Darcs, Apache Subversion or Mercurial) now being used will have become obsolete? In response, we have to build a second family of adaptors, called "loaders", to convert into a common, simple, maintainable data structure the information contained in this variety of versioning systems and pack formats.

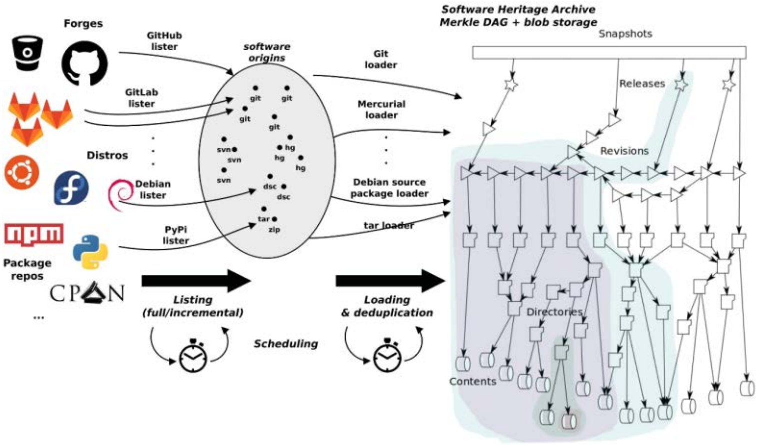
This data structure is a generalization of Merkle trees, which, invented more than forty years ago, figure in applications as varied as distributed version control systems, blockchains or distributed file systems (MERKLE 1987). This data structure has several advantages: each archived artifact is assigned an identification that can be verified independently (DI COSMO *et al*. 2019); the contents

[2] Founded in 1967, the National Institute for Research in Computer Science and Automation (INRIA) is a public establishment specialized in mathematics and computer science under the oversight of the Ministry of Higher Education, Research and Innovation and the Ministry of the Economy and Finances.

[3] https://www.softwareheritage.org/swhap

are “deduplicated” so as to considerably reduce the archive’s volume; and the resulting graph can be used to keep track of how various programs have reused the same source code. Figure 1 presents this approach’s architecture.

Figure 1: Architecture of Software Heritage’s harvesting system.



A universal mission

Beyond the technical complexity of the archival process, Software Heritage’s universal mission raises major questions about the strategy to be pursued. How to see to this nonprofit’s long-term viability? How to make sure that it remains at the service of everyone and not be privatized or fall under the control of private interests? How to locate all the source code written in the past few decades? How to maximize the chances for this precious collection’s perpetuity?

These questions have led Software Heritage to establish a few founding principles (ABRAMATIC *et al*. 2018, DI COSMO & ZACCHIROLI 2017): the systematic use of open software to build the organization’s infrastructure (so that its operation can be understood and, if necessary, replicated); the construction of a worldwide network of mirrors independent of the archive (since a large number of copies is the best protection against losses and attacks); the decision to set up an international nonprofit organization with several partners (so as to minimize the risks of having a single point of failure and see to it that Software Heritage will remain at the service of all). It is important to point out that persons like Jean-François Abramatic and Stefano Zacchiroli have had long careers devoted to the common good. For its mission, Software Heritage needs institutional legitimacy, and it must be capable of remaining open so as to muster a broad consensus. In this respect, the framework agreement signed between INRIA and UNESCO on 3 April 2017 was both a recognition of the importance of this mission and a major opportunity for establishing cooperation at the global level for accomplishing it.

A big step in this direction was the meeting of an international group of experts organized at UNESCO in November 2018. It resulted in the “Paris call for software source code as a heritage”.[4] Besides explaining why source code has become a major issue, this call detailed concrete recommendations for responding to this situation. One of them lends support to the efforts that started with Software Heritage for building an international infrastructure for storing source codes.

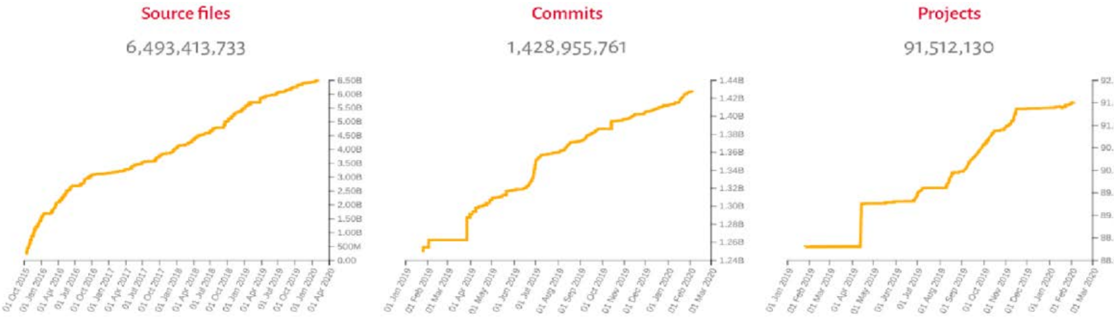
[4] https://en.unesco.org/foss/paris-call-software-source-code.

More than an archives: Past, present and future

Software Heritage is growing day after day. Although the largest proportion of the contents in its archives comes from the automated harvesting of Internet sites, pearls are starting to be turned up thanks to the patient work (which has come out of cooperation with Pisa University and UNESCO)[3] of recuperating major historical software programs.

Though far from exhaustive, this archive is already the largest corpus of source code available on the planet: more than 90 million from archived sources for more than 6 billion unique files of source code, each assigned an intrinsic identification based on cryptographic hashes (DI COSMO *et al*. 2018).

Figure 2:
Number of projects, source files and archived versions in Software Heritage (January 2020
Source: https://www.softwareheritage.org/archive



This unprecedented infrastructure has several purposes. One is, of course, to preserve for futures generations the source codes from the past that have marked the history of computer science and of the information society. But above all, we are trying to build a “very big telescope” for exploring the current evolution of the galaxy of software development in order to better understand and improve it so as to prepare a better technological future.

A strategic issue

The Software Heritage archive is already the largest source code collection on the planet, but the road ahead is still long. It is necessary to continue bringing together the scientific and technical skills needed for this project, not to mention the financial and human resources, in order to be able to preserve the memory of how technology and science have driven the digital revolution — at a time when we can still hope to obtain the software used since the start of this still short history of the information sciences. Even more: unrestricted access to the source code of publicly available software and to the information descriptive of the software’s development has become a matter of digital sovereignty for all nations at a time when software has definitely become a key component of all human activities.

The infrastructure built by Software Heritage and its universal approach are essential for addressing the issue of digital sovereignty but while preserving the principle of a pool of common goods, a typical characteristic of archives. It is, therefore, of utmost importance for institutions, industries, academics and stakeholders from society to realize the importance of this issue. France and Europe must rapidly work out a position and furnish the resources that Software Heritage needs to grow and last. By supporting the foundation of a non-profit international institution that will pursue this long-term mission, they will take a place alongside other international parties who have already made commitments.[5]

References

ABELSON H., SUSSMAN G. & SUSSMAN J. (1985) *Structure and Interpretation of Computer Programs* (Cambridge, MA: MIT Press), available at https://mitpress.mit.edu/sites/default/files/sicp/index.html.

ABRAMATIC J.F., DI COSMO R. & ZACCHIROLI S. (2018) "Building the universal archive of source code", *Commununicatons of the ACM*, 61(10), pp. 29-31, available at https://doi.org/10.1145/3183558.

DI COSMO R., GRUENPETER M. & ZACCHIROLI S. (September 2018), "Identifiers for digital objects: The case of software source code preservation", *Proceedings of the 15th International Conference on Digital Preservation*, iPRES 2018, Boston, USA, available at https://doi.org/10.17605/OSF.IO/KDE56.

DI COSMO R. & ZACCHIROLI S. (2017) "Software Heritage: Why and how to preserve software source code", *Proceedings of the 14th International Conference on Digital Preservation*, iPRES, September, available via https://hal.archives-ouvertes.fr/hal-01590958/file/ipres-2017-software-heritage.pdf.

DI COSMO R., GRUENPETER M. & ZACCHIROLI S. (2019) "Referencing source code artifacts: A separate concern in software citation", *Computing in Science & Engineering* 22(2), pp. 33-43, available at https://doi.org/10.1109/MCSE.2019.2963148.

GNU (2017) "General Public License", version 3 of 29 June available at https://www.gnu.org/licenses/gpl-3.0.html.

KNUTH D.E. (1984) "Literate programming", *The Computer Journal*, 27(2), pp. 97-111, available at https://doi.org/10.1093/comjnl/27.2.97

MERKLE R.C. (1987) "A digital signature based on a conventional encryption function" in C. POMERANCE (editor), *Advances in Cryptology — CRYPTO '87, A Conference on the Theory and Applications of Cryptographic Techniques* at Santa Barbara, California, USA, August 16-20, 1987, *pp.* 369-378, available at https://doi.org/10.1007/3-540-48184-2_32.

NOORDEN R.V., MAHER B. & NUZZO R. (2014) "The top 100 papers: *Nature* explores the most-cited research of all time", *Nature*, 514, 30 October, pp. 550-553, available via https://doi.org/10.1038/514550a.

SHUSTEK L.J. (2006) "What should we collect to preserve the history of software?", *IEEE Annals of the History of Computing*, 28(4), pp. 110-112, available at https://doi.org/10.1109/MAHC.2006.78.

[5] For more on our project, *cf.*: www.softwareheritage.org, annex.softwareheritage.org and wiki.softwareheritage.org
The source code in Software Heritage can be easily explored on archive.softwareheritage.org.