Ten years of research on open source software

Di Cosmo, Leroy, Treinen, Vouillon et al
*Managing the complexity of large free and open source package-based software distributions, ASE 2006*

Abate, Boender, Di Cosmo, Zacchirol
*Strong Dependencies between Software Components, ESEM 2009*

Di Cosmo and J. Vouillon.
*On software component co-installability, ESEC/FSE 2011*

Abate, Di Cosmo, Treinen, Zacchirol
*Learning from the Future of Component Repositories, CBSE 2012*

Vouillon, Dogguy, Di Cosmo.
*Easing software component repository evolution. ICSE 2014*

Abate, Di Cosmo, Gesbert, Le Fessant, Treinen, and Zacchirol.
*Mining component repositories for installability issues, MSR 2015*

Claes, Mens, Di Cosmo, and Vouillon.
*A historical analysis of debian package incompatibilities, MSR 2015*

**Tools**

- Cudf library: http://gforge.inria.fr/projects/cudf/
- Dose library: http://gforge.inria.fr/projects/dose/
- Coinst suite: http://coinst.irill.org
- Debian QA: http://qa.debian.org/dose
A recurring pattern

- identify a real world problem whose solution requires a research effort
- work hard to find a solution
- implement a tool, validate it on real world cases
- publish a research article
- foster adoption (the hardest part!)

In a picture

Under the hood

Question:

What were the technical prerequisites that made this work possible?
Technical and legal enablers

Availability
- all the *(history of)* Debian packages (since 2005)
- no *technical* restrictions
- no *legal* restrictions on *content* or *metadata*

Traceability
Debian packages have
- *unique identifier*
- *reference central repository*

Uniformity
Debian packages: a reference catalog
- *uniform metadata structure*
- *uniform naming and versioning schema*

These are all essential features
for *reproducibility* and for *preservation*…
… we need them for *all* software!
Software is everywhere

At the heart of our society

- communication, entertainment
- administration, finance
- health, energy, transportation
- education, research, politics
- ...

Knowledge enabler

- *Key mediator* for accessing *all* information
- *Essential component* of modern scientific research

Software embodies

our collective *Knowledge and Cultural Heritage*
Software is fragile

- Inconsiderate and/or malicious code loss (e.g., Code Spaces)
- Business-driven code loss (e.g., Gitorious, Google Code)
- For obsolete code: physical media decay (data rot)

If a website disappears you go to the Internet Archive...

...where do you go if (a repository on) GitHub goes away?
Software is spread all around

many disparate development platforms
a myriad places where distribution may happen
projects tend to migrate from one place to the other over time

... where can we find, track and search all the source code?
Software is missing its own Research Infrastructure

Photo: ALMA(ESO/NAOJ/NRAO), R. Hills

A wealth of software research on crucial issues…

- safety, security; test, verification, proof;
- software engineering, software evolution;
- empirical and big data studies;

If you study the stars, you go to Atacama…

… where is the very large telescope of source code?
Collect, organise, preserve and share the source code of all the software that lies at the heart of our culture and our society.

Preserving the past, enhancing the present, preparing the future.
Software Source Code is different

“Programs must be written for people to read, and only incidentally for machines to execute.” Harold Abelson, *Structure and Interpretation of Computer Programs*

### Distinguishing features

- executable and *human readable* knowledge (an *all time new*)
  - even hardware is… software! (VHDL, FPGA, …)
  - *text files are forever*
- naturally *evolves* over time
  - the *development history* is key to its *understanding*
- complex: large *web of dependencies*, millions of SLOCs

### In a word

- *software is not just another* sequence of bits
- a *software archive is not just another* digital archive
We are working on the foundations

one infrastructure to build them all

- Mankind’s memory
- Long term preservation
- Unique reference
- Software Wikipedia

Cultural Heritage

- Reference repository
- Provenance
- Certification
- Security

Industry

- Reproducibility
- Traceability
- Open Access
- Software studies

Research

- Universal SourceBook
- Reference examples
- Enriched source code
- Code documentation

Education

Software Heritage
Supporting more accessible and reproducible science

A global library referencing all software used in all research fields

- completes the infrastructure for Open Access in science
- provides intrinsic persistent identifiers needed for scientific reproducibility
- enables large scale, verifiable software studies
The Knowledge Conservancy Magic Triangle

**Legenda (links are important!)**

- **articles**: ArXiv, HAL, …
- **data**: Zenodo, …
- **software**: *Software Heritage* to the rescue
**Repeatable Software Studies**

- vulnerability detection
- dependency analysis
- pattern elicitation
- study of the development graph
- ... the sky is the limit

**Prerequisites**

clean, evolvable data and metadata model
### Availability

- *all the history of all the software*
- no restrictions (technical, legal, … ) on *content* or *metadata*

### Traceability

- *unique* identifiers: *one* name for each object
- *persistent* and *intrinsic* identifiers: no middle man, no dangling pointers!

### Uniformity

- one *standard* metadata structure, *irrespective of the origins*
- *uniform* naming *schema*

---

here are some bits from our drawing board
Free and Open Source Software is crucial

D. Rosenthal, EUDAT, 9/2014

You have to do [digital preservation] with open-source software; closed-source preservation has the same fatal "just trust me" aspect that closed-source encryption suffer from.

design decision

Software Heritage will:

- provide full details on its architecture
- make available all the source code used
- use open standards
- encourage a collaborative development process
- unleash and leverage the power of the community
Web links are not permanent (even *permalinks*).


*Users should beware that there is no general guarantee that a URL which at one time points to a given object continues to do so, and does not even at some later time point to a different object due to the movement of objects on servers.*

The Decay and Failures of URL References

*half life of web references is 4 years*

*Diomidis Spinellis, CACM 2003*

**design decision**

Software Heritage will:

- provide *intrinsic* resource identifiers
- *avoid* volatile identifiers like DOI or URLs
Replication is the key

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.

design decision

Software Heritage will:

- provide easy means for making copies
- encourage the growth of a mirror network
  - using a variety of technologies
  - spanning multiple continents
  - under diverse control structures
    - no single decisional point of failure!
      (remember Google code, Gitorious, …)
Why us? Because the Source Code is our DNA!

it is at the heart of our work

- we *write* software
- we *read and reuse* software
- we *distribute* software
- we *understand* how software works

Bottomline

it is our *duty* and our *privilege* to take care of Software preservation
The team

- Roberto Di Cosmo
- Stefano Zacchirolì
- Nicolas Dandrimont
- Antoine Dumont
- and Guillaume, Quentin, Jordi

Scientific advisors

- Serge Abiteboul
- Jean-François Abramatic
- Gerard Berry

... and all of Inria’s support
Where we are today: technically

Data model: full development history, VCS-independent

- the biggest "Git" graph in the world?

Our sources

- GitHub — all public repositories, as of April 2016
- Debian — daily snapshots of all suites since 2005–2015
- GNU — all historical releases up to August 2015
- Gitorious — retrieved full mirror from Archive Team
- Google Code — retrieved full mirror from Google

Some numbers

- 22 million repositories ingested (10M next in line)
- 600 million commits
- 2.2 billion directories
- 2.7 billion unique files / 120 TB of (cmpd) raw source code
What’s special

**Uniform data model**
- superset of *git* : ambition to *cover all VCS*
  - contents, directories, revisions, releases, origins, …

**Massive deduplication**
- the biggest git-like graph in the world right now
  - did you know? the original GPLv2 licence
    - appears with more than 500 different file names
    - including LICENSE-2 and FullSync.txt ~ :-)

**Provenance tracking**
- know *where* we found *what, when*
- essential for *traceability*
Where we are today

Inria as initiator
- funds the *bootstrap phase* of Software Heritage
- going global: an *open, nonprofit* organisation

Come in, we’re open: everybody is needed!
- **researchers** scientific challenges
- **developers** Software Heritage is itself Open Source!
- **archivists** find the many source code repositories
- **partners** contribute to the effort

We are happy to welcome our *first* partners!
- Microsoft: leading software company, CodePlex, Azure
- DANS (Royal Academy of the Arts and Sciences): sustained access to research data
Come in, we’re open

Software Heritage working groups
https://wiki.softwareheritage.org

Resources for distributed storage
share storage/compute nodes for research use

Adoption
- help connecting Software Heritage with everyday’s work
- spread its use across research communities

Research
metadata, linked data, big data, distribution/replication, search, …

Our forge opens today!
https://forge.softwareheritage.org/
Some planned working groups

Source Discovery and Ingestion (SODI)
- API for listing the contents of a source
- mechanisms for discovering new sources

Scientific APIs (SAPI)
- monitor needs of the research community
- API for accessing Software Heritage data as a research corpus

Open Access and Data (OPAD)
- develop common standards for cross referencing artefacts
- monitor and evaluate existing and forecoming approaches to unique persistent identifiers
- raise awareness, and foster broad adoption of the Software Heritage’s software identifiers
Conclusion

Software Heritage is

- a revolutionary *reference archive* of all software ever written
- a unique *complement* for *development platforms*
- an international, open, nonprofit, *mutualized infrastructure*

*we need your help to make it happen*

Time to visit

https://www.softwareheritage.org!

Questions?

Keeping in contact

*mailing list*: swh-science@inria.fr

https://sympa.inria.fr/sympa/info/swh-science