Preserving Software
Challenges and opportunities for reproductibility

Roberto Di Cosmo

Journée Reproductibilité
roberto@dicosmo.org

9 Décembre 2014
Reproducibility (Wikipedia)
the ability of an entire experiment or study to be reproduced, either by the researcher or by someone else working independently. It is one of the main principles of the scientific method.

Reproducibility in the digital age
For an experiment involving software, we need

- open access to the scientific article describing it
- open data sets used in the experiment
- source code of all the components
- environment of execution
- stable references between all this

The first two items are already widely discussed!
Some people claim that having (all) the source of the code used in an experiment is *not worth the effort* \(^1\).
Sure, diversity *is* important, but consider that:

- source code is like the proof used in a theorem: can we really accept *Fermat statements* like “the details are omitted due to lack of space”?
- and even more so when the complexity of modern systems makes even the simplest experiment depend on a wealth of components and configuration options?
- having access to all the source code is not just necessary to *reproduce*, it is also useful to *evolve and modify*, to *build new experiments* from the old ones

\(^1\)“Replicability is not Reproducibility: Nor is it Good Science”, Chris Drummond, ICML 2009
Digital Preservation (Wikipedia)

In library and archival science, digital preservation is a *formal endeavor* to ensure that digital information of continuing value *remains accessible and usable*.

Digital Preservation for Reproducibility

(Digital) preservation is the *unstated assumption* underlying reproducibility efforts for all scientific experiments:

we cannot reproduce an experiment whose description has been lost!

What is a *description*?

In our modern world, this comprises *articles, data* and, yes, *software*!
Software is Science’s cornerstone

Software is an essential component of modern scientific research

Deep knowledge from mathematics, physics, chemistry, biology, medicine, finance and social sciences is now inextricably embodied into complex software systems, at a level of detail that goes way beyond that of the usual scientific publications.

Top 100 papers (Nature, October 2014)

[...] the vast majority describe experimental methods or software that have become essential in their fields.

http://www.nature.com/news/the-top-100-papers-1.16224

Bottomline: Software is Knowledge that needs to be preserved!
Like all digital information, software is *fragile*

### Causes of information loss

<table>
<thead>
<tr>
<th>Category</th>
<th>Causes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Human</strong></td>
<td>accidental or malicious deletion, ...</td>
</tr>
<tr>
<td><strong>Storage Media, Practices, Systems</strong></td>
<td>disaster events, corruption, damage, wear and tear, aging ...</td>
</tr>
<tr>
<td><strong>Logical Format or Migration</strong></td>
<td>Inability to Access, Read, Interpret, Validate, or Use information</td>
</tr>
<tr>
<td></td>
<td>Loss of the necessary <strong>software tools</strong></td>
</tr>
<tr>
<td></td>
<td>Loss or damage to <strong>references</strong> to associated information</td>
</tr>
<tr>
<td><strong>Encryption</strong></td>
<td>Lost access keys, decryption devices</td>
</tr>
<tr>
<td><strong>Authenticity</strong></td>
<td>Failure to identify the <strong>intended</strong> versions</td>
</tr>
<tr>
<td><strong>External Service Providers</strong></td>
<td>Out of business, high exit cost, ...</td>
</tr>
</tbody>
</table>

An example is worth a thousand words...
Inconsiderate loss of code: Y2K

Y2K : The Year 2000 Bug Crisis

The announced disasters did not occur, and we’ll never know if it’s because of the billions spent on fixing the bug, but ...

An Inconvenient Truth

... this bug uncovered the astonishing fact that, in 1999, an estimated 40% of companies had either lost, or thrown away the original source code for their systems!
Malicious loss of code: R.I.P. CodeSpaces

THE DEEP END
By Paul Venezia | Follow

Murder in the Amazon cloud

The demise of Code Spaces at the hands of an attacker shows that, in the cloud, off-site backups and separation of services could be key to survival

InfoWorld | Jun 23, 2014

Code Spaces was a company that offered developers source code repositories and project management services using Git or Subversion, among other options. It had been going for seven years, and it had no shortage of customers. But it's all over now -- the company was essentially murdered by an attacker.

Yes, for seven years all seemed good and well!

No, they did not recover the data.

MORE LIKE THIS

Prepare yourself for high-stakes cyber ransom

6 lessons learned about the scariest security threats

Security-vendor snake oil: 7 promises that don’t deliver

Roberto Di Cosmo | Preserving Software | 11/27
Project Hosting on Google Code provides a free collaborative development environment for open source projects. Each project comes with its own member controls, Subversion/Mercurial/Git repository, issue tracker, wiki pages, and downloads service.

Downloads were implemented by Project Hosting on Google Code to enable open source projects to make their files available for public download. Unfortunately, downloads have become a source of abuse with a significant increase in incidents recently. Due to this increasing misuse of the service and a desire to keep our community safe and secure, we are deprecating downloads.

Starting today, existing projects that do not have any downloads and all new projects will not have the ability to create downloads. Existing projects with downloads will see no visible changes until January 14, 2014 and will no longer have the ability to create new downloads starting on January 15, 2014. All existing downloads in these projects will continue to be accessible for the foreseeable future.

If your project is using downloads to host and distribute files and has a need to periodically create new downloads, we recommend you move your downloads to an alternate service like Google Drive before January 15, 2014. If you choose to move your files to Google Drive, check out our help article.

By Google Project Hosting
Disruption of the web of reference: our Gforge

[siteadmin-Bugs][#17468] *Urls of release files has silently changed*

A noreply

---

siteadmin-**Bugs** [#17468] was **changed** at 2014-05-21 11:11 by Vincent Lefèvre.
You can respond by visiting:
https://gforge.inria.fr/tracker/?func=detail&aid=468&group_id1

**Status:** Open
**Priority:** 3
**Submitted By:** Roberto Di Cosmo (robertodicisco)
**Assigned to:** Nobody (None)
**Summary:** *Urls of release files has silently changed*
**Category:** génant
**Group:** None
**Resolution:** None

**Initial Comment:**
The *url of release files has silently changed*; for example, the original release file

https://gforge.inria.fr/frs/download.php/file/31910/cudf-0.6.3.tar.gz

now gives an empty file when downloading it, while the actual *url changed* to

https://gforge.inria.fr/frs/download.php/31910/cudf-0.6.3.tar.gz

There are surely good reasons for this, but I would like to stress the fact that we *need* to be able to rely on permanent URLs for releasing our software... these *urls end up embedded in other tools and software, and changing them is a source of unneeded problems.*

---

**Fixed, adding a redirection, by the Gforge team in 1 day!**

Not always that lucky, though ...
Disruption of the web of reference: an old problem

URLs used in articles do 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
- deep path hierarchies are linked to increased URL failures


Similar findings in
Preservation of digital information is on the rise

A wealth of initiatives around us

| generalist | the Web archive at archive.org; Digital Preservation Coalition (UK); National Digital Information Infrastructure and Preservation Program (NDIIPP, USA); ... |
| culture   | books, music, video: http://www.nationalarchives.gov.uk (UK); INA (FR); ... |
| social networks | Twitter is archived by the Library of Congress! |
| libraries and scholarly work | ArXiv; Digital Preservation Network http://www.dpn.org/; ... |
| scientific data | CINES (FR); Zenodo/OpenAire (CERN); ... |

What about the software?
Software is the mediator for our digital culture

*Absent an ability to correctly interpret digital information, we are left with files full of “rotting bits” that are of no value.*

Vinton G. Cerf


If we do not preserve software, **digital preservation is futile!**

And yet, up to now...

software is *largely ignored* as an object of preservation...

computer scientists are *mostly absent* in the preservation landscape!
Unlike for books or movies, there is a big difference between *using* and *understanding* a piece of software.

**Using software**
Requires an executable, and access to the *execution environment*

**Understanding software**
Requires access to the *source code*:

> The source code for a work means the preferred form of the work for making modifications to it.

— GNU General Public Licence, version 2

For reproducibility, we need *both*!
Preserving *software* is *more complex* than archiving books or scientific articles.

**Interdependencies** a program relies on other software (libraries, compilers, development tools, runtime systems, etc.) as well as specific hardware components (or equivalent virtual machines) to be executed; so does our understanding of its functioning.

**Evolution** software is a *live object*: its detailed history contains key knowledge that cannot be reconstructed by only looking at an individual snapshot of the software source code.

To preserve *software* it is *not enough* to mimic processes that were intended to archive books, scientific articles or data.
Software Preservation: a Unique Opportunity

The half empty glass point of view
Nobody cares about software...
Computer scientists are absent from the preservation landscape...

Nobody loves us...

The half full glass point of view
Preserving software is a highly challenging task...
... it requires Computer Scientists in the loop

Luckily nobody cared!

Let’s do it right, ... let’s do it now!
The Knowledge Conservancy Magic Triangle

- **Open Access Repositories**
- **Open Data Sets Repositories**
- **Open Source Repositories**

**Articles**  ArXiv, HAL?, ...
**Data**  Zenodo? (OpenAire)
**Software**  coming soon from Inria!
Replication is the key

...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.

Thomas Jefferson, February 18, 1791

recommendation

our preferred platform(s) should:

- provide easy means for making copies
- encourage the growth of a mirror network (like ArXiv did)
Free an Open Source Software is crucial

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 (and cloud storage) suffer from.

D. Rosenthal, EUDAT, 9/2014

recommendation

our preferred platform(s) should:

- provide full details on their architecture
- make available all the source code used
- use open standards
- encourage a collaborative development process

Unfortunately, this is not (yet?) the case for HAL or Zenodo
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.*


**recommendation**

Our preferred platform(s) should:

- provide *intrinsic* resource identifiers
- *avoid* intermediate index approaches like DOI
Conclusions

- Long term preservation is the *unspoken assumption* of all scientific reproducibility efforts.
- We need to preserve scientific articles, scientific data *and* software (magic triangle).
- Digital preservation is on the rise, mostly thanks to librarians, with almost no computer scientists involved.
- Software preservation is *not yet there* and is in dire need of attention.
- We have a great opportunity to seize… let’s do it!