La préservation des logiciels: défis et opportunités pour la reproductibilité en science et technologie

Roberto Di Cosmo

roberto@dicosmo.org

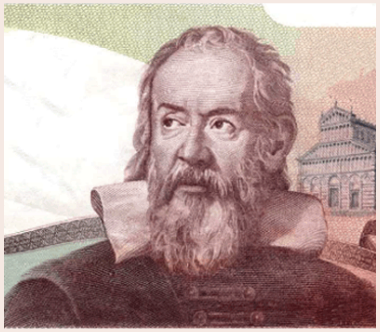
3 Décembre 2015

Software Heritage

Outline

How we built our scientific knowledge

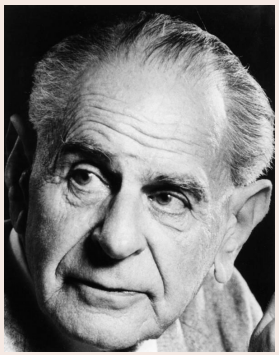
The experimental method



- make an *observation*
- formulate an *hypothesis*
- set up an experiment
- formulate a *theory*

And then we reproduce and verify.

Reproducibility is the key



non-reproducible single occurrences are of no significance to science

Karl Popper, The Logic of Scientific Discovery, 1934

Reproducibility, today

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.

Why we want it

- foundation of the scientific method
- accelerator of research: allows to build upon previous work
- visibility: reproducible results are cited more often
- transparency of results eases acceptance
- necessary for industrial transfer

reproducibility is *the essence* of *industry*!

Outline

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

Remark

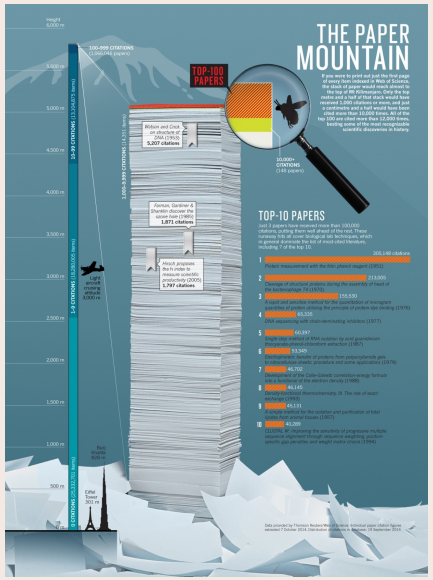
The first two items are already widely discussed!

Software is Knowledge

Software is *an essential component* of modern scientific research

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

Top 100 papers (Nature, October 2014) `http://www.nature.com/news/the-top-100-papers-1.16224`



Use the Source, Luke!

Some people claim that having (all) the source of the code used in an experiment is *not worth the effort* (see “Replicability is not Reproducibility: Nor is it Good Science”, Chris Drummond, ICML 2009)

Use the Source, Luke!

Some people claim that having (all) the source of the code used in an experiment is *not worth the effort* (see "Replicability is not Reproducibility: Nor is it Good Science", Chris Drummond, ICML 2009)

Sure, diversity *is* important, but:

- 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"?
- modern complex systems makes even the simplest experiment depend on a wealth of components and configuration options
- 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

Outline

Software and reproducibility

A fundamental question

How are we doing, regarding reproducibility, in *Software*?

The case of Computer Systems Research

A field with Computer experts ... we have high expectations!
Christian Collberg set out to check them.

Measuring Reproducibility in Computer Systems Research

Long and detailed technical report, March 2014
`http://reproducibility.cs.arizona.edu/v1/tr.pdf`

Collberg's report from the trenches

Analysis of 613 papers

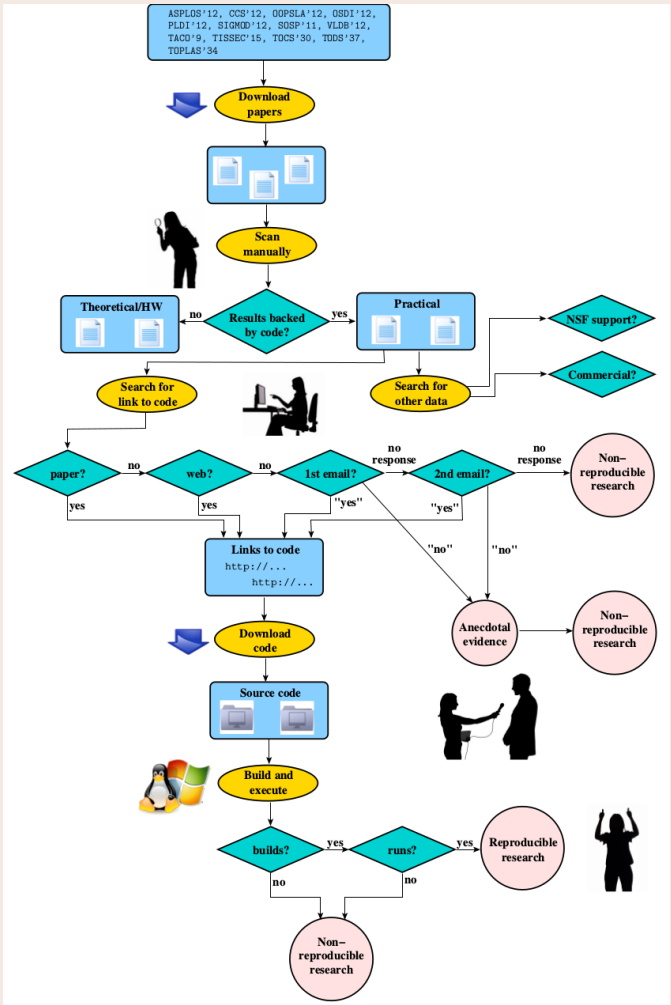
- 8 ACM conferences: ASPLOS'12, CCS'12, OOPSLA'12, OSDI'12, PLDI'12, SIGMOD'12, SOSP'11, VLDB'12
- 5 journals: TACO'9, TISSEC'15, TOCS'30, TODS'37, TOPLAS'34

all very practical oriented

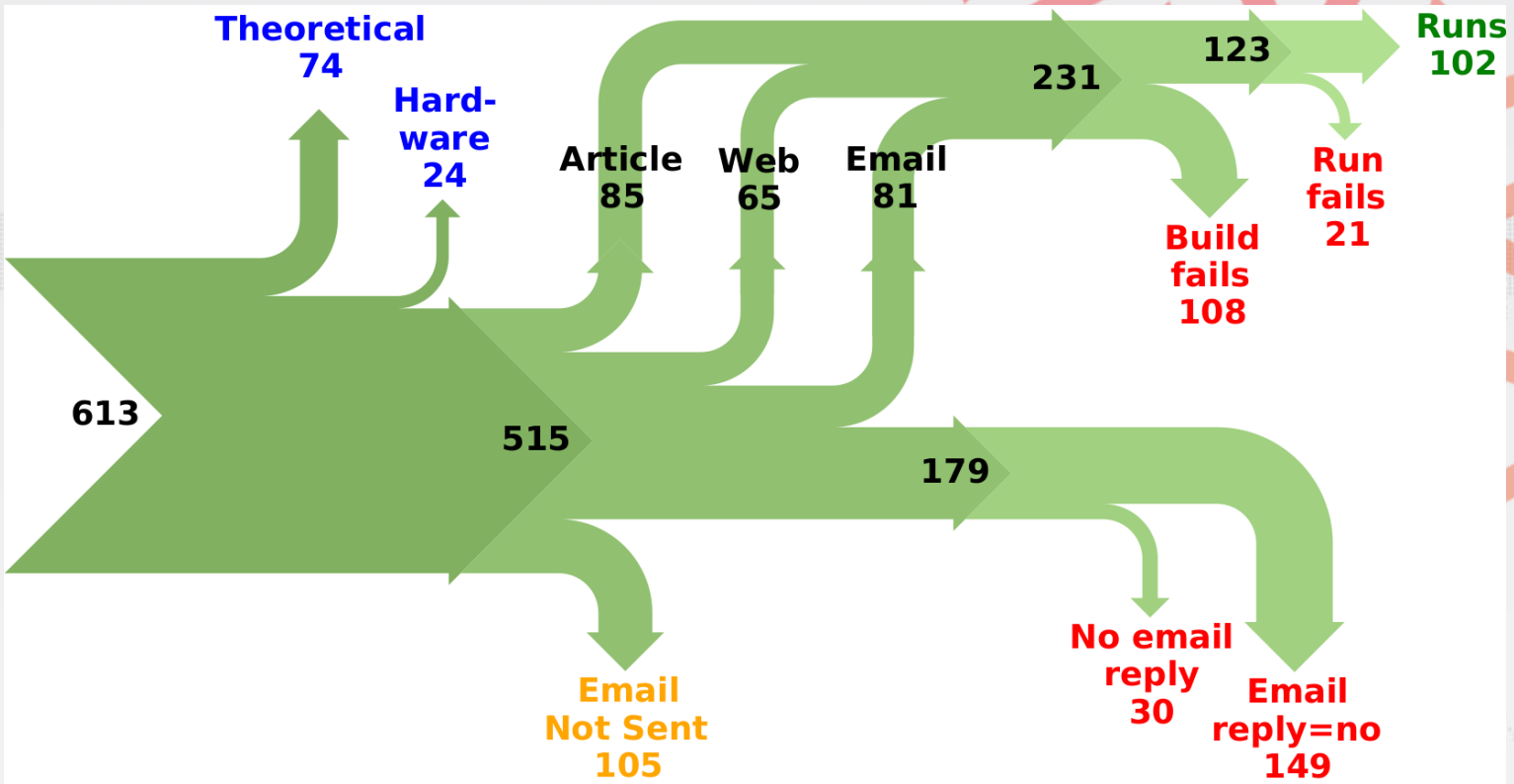
The basic question

can we get the code to build and run?

The workflow



The result



Even if these numbers can be debated …

… that's a whopping 81% of non reproducible works!

The reasons (or, “the dog ate my program”)

Many issues, nice anectotes, and it finally boils down to

- *Availability*
- *Traceability*
- Environment
- Automation (do *you* use continuous integration?)
- Documentation
- Understanding (including Open Source)

The reasons (or, “the dog ate my program”)

Many issues, nice anectotes, and it finally boils down to

- *Availability*
- *Traceability*
- Environment
- Automation (do *you* use continuous integration?)
- Documentation
- Understanding (including Open Source)

The first two are important *software preservation issues*

Yes, code is fragile:

it can be destroyed, and we can loose trace of it

Outline

Like all digital information, software is fragile

An example is worth a thousand words…

let's see a few

Inconsiderate or malicious loss of code

The Year 2000 Bug … uncovered an inconvenient truth

in 1999, an estimated 40% of companies had either *lost*, or thrown away the original source code for their systems!

CodeSpaces: source code hosting, 2007-2014

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

Yes, for *seven years* all seemed ok.

Inconsiderate or malicious loss of code

The Year 2000 Bug … uncovered an inconvenient truth

in 1999, an estimated 40% of companies had either *lost*, or thrown away the original source code for their systems!

CodeSpaces: source code hosting, 2007-2014

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

Yes, for *seven years* all seemed ok.
No, they did not recover the data.

Business-driven loss of code support: Google

A Change to Google Code Download Service

Posted: Monday, May 20, 2013

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

Business-driven loss of code support: Google, cont'd.

Posted: Thursday, March 12, 2015

+1 377 Tweet 1,210 Like 404

When we started the Google Code project hosting service in 2006, the world of project hosting was limited. We were worried about reliability and stagnation, so we took action by giving the open source community another option to choose from. Since then, we've seen a wide variety of better project hosting services such as GitHub and Bitbucket bloom. Many projects moved away from Google Code to those other systems. To meet developers where they are, we ourselves migrated nearly a thousand of our own open source projects from Google Code to GitHub.

As developers migrated away from Google Code, a growing share of the remaining projects were spam or abuse. Lately, the administrative load has consisted almost exclusively of abuse management. After profiling non-abusive activity on Google Code, it has become clear to us that the service simply isn't needed anymore.

Beginning today, we have disabled new project creation on Google Code. We will be shutting down the service about 10 months from now on January 25th, 2016. Below, we provide links to migration tools designed to help you move your projects off of Google Code. We will also make ourselves available over the next three months to those projects that need help migrating from Google Code to other hosts.

- March 12, 2015 - New project creation disabled.
- August 24, 2015 - The site goes read-only. You can still checkout/view project source, issues, and wikis.
- January 25, 2016 - The project hosting service is closed. You will be able to download a tarball of project source, issues, and wikis. These tarballs will be available throughout the rest of 2016.

Google will continue to provide Git and Gerrit hosting for certain projects like Android and Chrome. We will also continue maintaining our mirrors of projects like Eclipse, kernel.org and others.

Business-driven loss of code support: Gitorious

From: Rolf Bjaanes <rolf@gitorious.org>
To: zack@upsilon.cc
Subject: Gitorious.org is dead, long live Gitorious.org
Message-Id: <30589491.20150416155909.552fdc4d164

Hi zacchiro,

I'm Rolf Bjaanes, CEO of Gitorious, and you are receiving this email because you have a user on gitorious.org. As you may know, Gitorious was acquired by GitLab [1] about a month ago *(NDLR: 3/3/2015)*, and we announced that Gitorious.org would be shutting down at the end of May, 2015.
... Rolf

Disruption of the *web of reference*

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
T. Berners-Lee et al. Uniform Resource Locators. RFC 1738.

Disruption of the *web of reference*

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
T. Berners-Lee et al. Uniform Resource Locators. RFC 1738.

404

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 D. Spinellis. The Decay and Failures of URL References.
  Communications of the ACM, 46(1):71-77, January 2003.

Disruption of the web of reference: our Gforge

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

siteadmin-bugs@gforge.inria.fr via dicosmo.org 21 mai

À noreply

anglais › français Traduire le message Désactiver pou

siteadmin-Bugs [#17468] was changed at 2014-05-21 11:11 by Vincent Lefèvre
You can respond by visiting:
https://gforge.inria.fr/tracker/?funcÞtail&atid 0&aid 468&group_id1

Status: Open
Priority: 3
Submitted By: Roberto Di Cosmo (robertodicosmo)
Assigned to: Nobody (None)
Summary: Urls of release files has silently changed
Category: gênant
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.

Disruption of the web of reference: our Gforge

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

siteadmin-bugs@gforge.inria.fr via dicosmo.org 21 mai

À noreply

anglais › français Traduire le message Désactiver pou

siteadmin-Bugs [#17468] was changed at 2014-05-21 11:11 by Vincent Lefèvre
You can respond by visiting:
https://gforge.inria.fr/tracker/?funcÞtail&atid 0&aid 468&group_id1

Status: Open
Priority: 3
Submitted By: Roberto Di Cosmo (robertodicosmo)
Assigned to: Nobody (None)
Summary: Urls of release files has silently changed
Category: gênant
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* this one was fixed!

Disruption of the web of reference: our Gforge

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

siteadmin-bugs@gforge.inria.fr via dicosmo.org 21 mai
À noreply

anglais › français Traduire le message Désactiver pou

siteadmin-Bugs [#17468] was changed at 2014-05-21 11:11 by Vincent Lefèvre
You can respond by visiting:
https://gforge.inria.fr/tracker/?funcÞtail&atid 0&aid 468&group_id1

Status: Open
Priority: 3
Submitted By: Roberto Di Cosmo (robertodicosmo)
Assigned to: Nobody (None)
Summary: Urls of release files has silently changed
Category: gênant
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* this one was fixed!

Not always that lucky, though …

Outline

Software Heritage

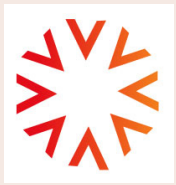
PRESERVING TECHNICAL KNOWLEDGE

Our mission

Collect, *organise*, *preserve* and *share all the software* that lies at the heart of our culture and our society.

Our mission in our logo

Collect



We collect software, all of it.

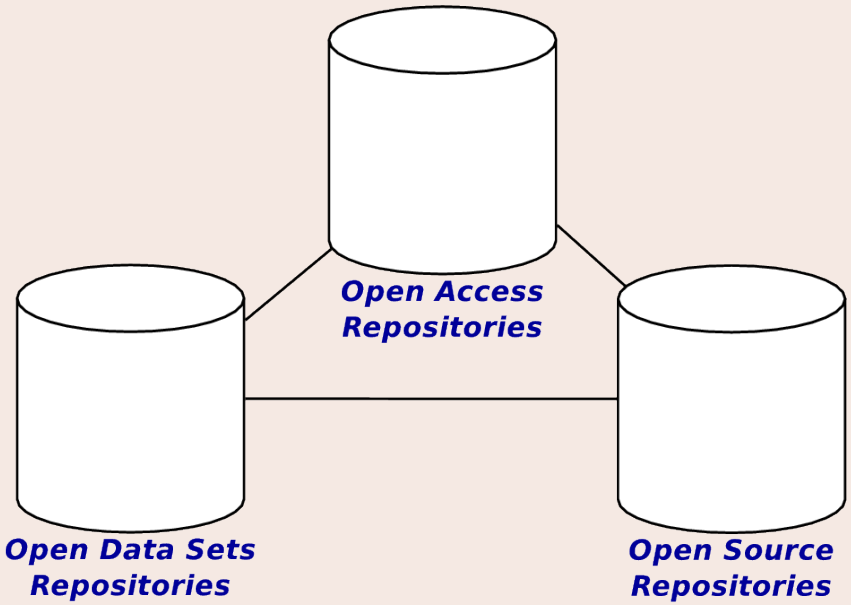
Preserve

We preserve software, because it embodies our technical and scientific knowledge.

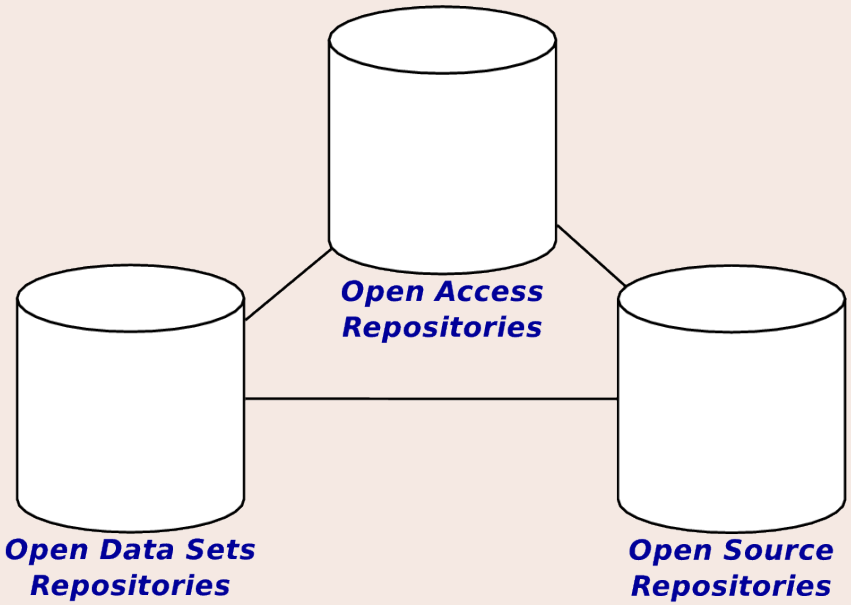
Share

We will index, organise, and make widely accessible all the Software we collect

The Knowledge Conservancy Magic Triangle



The Knowledge Conservancy Magic Triangle



Legenda (links are important!)

- articles: ArXiv, HAL, ...
- data: Zenodo, ...
- software: *Software Heritage* to the rescue

Outline

Free an 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 (and cloud storage) suffer from.

Free an 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 (and cloud storage) suffer from.

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

Web links *are not* permanent (even *permalinks*)

T. Berners-Lee et al. Uniform Resource Locators. RFC 1738.

> *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.*

Web links *are not* permanent (even *permalinks*)

T. Berners-Lee et al. Uniform Resource Locators. RFC 1738.

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

Web links *are not* permanent (even *permalinks*)

T. Berners-Lee et al. Uniform Resource Locators. RFC 1738.

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

recommendation

our preferred platform(s) should:

- 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.

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.

recommendation

our preferred platform(s) should:

- provide easy means for making copies
- encourage the growth of a mirror network (like ArXiv did)

Caring for the long term

not just a project

projects have limited time-frame

not commercial

business interests come and go

a shared concern

- cultural heritage
- scientific infrastructure
- industrial infrastructure

Unix philosophy

do *one* thing, do it *well*

Outline

Need your help

make it easy to integrate

- in development workflow
- in publishing workflow

Need your help

make it easy to integrate

- in development workflow
- in publishing workflow

make it ok to integrate, from the legal point of view

- make licences explicit
- make licences of dependencies explicit

Need your help

make it easy to integrate

- in development workflow
- in publishing workflow

make it ok to integrate, from the legal point of view

- make licences explicit
- make licences of dependencies explicit

make it sustainable

- support/sponsorship
- open process
- collaboration

Focus on the legal issues

a plurality of concerns

Who owns the rights to your research?

- articles, data, software
- too often forgotten: metadata
  - Software Track in Science of Computer Programming, 2015
  - You own the software, but who owns the metadata?

we need to recover our rigths

- it is possible!
  - compulsory exclusive copyright transfer for free
    - is *illegal* in France (art L. 131-4 of CPI)
    - is debatable in all jurisdictions
- see Free Scientific Publication
- paying the editors (OpenAire) is *not a solution*

Outline

Questions?

Pour rester en contact

mailing list: swh-science@inria.fr

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