Research Software Hackaton

Introduction and highlights

Roberto Di Cosmo

Online material: http://bit.ly/reswhack

October 15th, 2019

Software Heritage

THE GREAT LIBRARY OF SOURCE CODE

Outline

1. Introduction
2. Academia's evolving practice
3. Connecting communities
4. Challenges
5. Moving forward

Short Bio: Roberto Di Cosmo

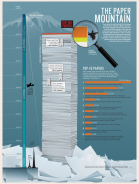
Computer Science professor in Paris, now working at INRIA

- *30 years* of research (Theor. CS, Programming, Software Engineering, Erdos #: 3)
- *20 years* of Free and Open Source Software
- *10 years* building and directing structures for the common good

1999	*DemoLinux* – first live GNU/Linux distro
2007	*Free Software Thematic Group* 150 members 40 projects 200Me
2008	*Mancoosi project* www.mancoosi.org
2010	*IRILL* www.irill.org
2015	*Software Heritage* at INRIA
2018	*National Committee for Open Science*, France

Why we are here

Software is everywhere in modern research

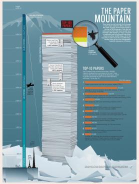


[...] software [...] essential in their fields.
Top 100 papers (Nature, 2014)

Sometimes, if you dont have the software, you dont have the data
Christine Borgman, Paris, 2018

Why we are here

Software is everywhere in modern research



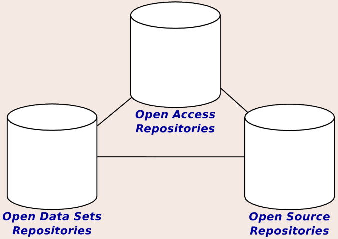
[…] software […] essential in their fields.

Top 100 papers (Nature, 2014)

Sometimes, if you dont have the software, you dont have the data

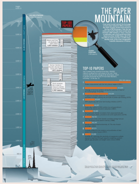
Christine Borgman, Paris, 2018

Open Science: three pillars



Why we are here

Software is everywhere in modern research



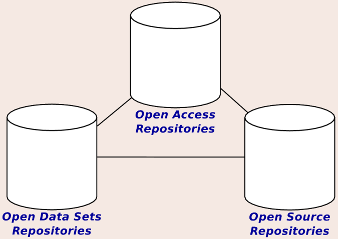
[...] software [...] essential in their fields.

Top 100 papers (Nature, 2014)

Sometimes, if you dont have the software, you dont have the data

Christine Borgman, Paris, 2018

Open Science: three pillars



Nota bene

The links in the picture are essential

The knowledge is in the source code!

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

GPL Licence

The knowledge is in the source code!

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

GPL Licence

Hello World

The knowledge is in the source code!

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

Hello World

Program (excerpt of binary)

```
4004e6: 55
4004e7: 48 89 e5
4004ea: bf 84 05 40 00
4004ef: b8 00 00 00 00
4004f4: e8 c7 fe ff ff
4004f9: 90
4004fa: 5d
4004fb: c3
```

The knowledge is in the source code!

> *“The source code for a work means the preferred form of the work for making modifications to it."*
> *GPL Licence*

Hello World

Program (excerpt of binary)

```
4004e6: 55
4004e7: 48 89 e5
4004ea: bf 84 05 40 00
4004ef: b8 00 00 00 00
4004f4: e8 c7 fe ff ff
4004f9: 90
4004fa: 5d
4004fb: c3
```

Program (source code)

```
/* Hello World program */

#include<stdio.h>

void main()
{
    printf("Hello World");
}
```

Source code is *special*

Executable and *human readable* knowledge — copyright law

“Programs must be written for people to read, and only incidentally for machines to execute.”

Harold Abelson

Source code is *special*

Executable and *human readable* knowledge — copyright law

"Programs must be written for people to read, and only incidentally for machines to execute."

Harold Abelson

Software *evolves* over time

- projects may last decades
- the *development history* is key to its *understanding*

Source code is *special*

Executable and *human readable* knowledge copyright law

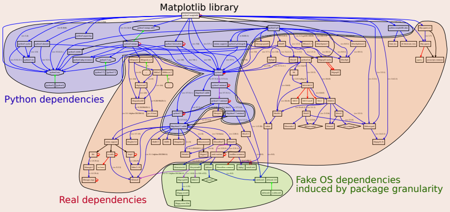
> *"Programs must be written for people to read, and only incidentally for machines to execute."*
>
> Harold Abelson

Software *evolves* over time

- projects may last decades
- the *development history* is key to its *understanding*

Complexity

- *millions* of lines of code
- large *web of dependencies*
  - easy to break, difficult to maintain
- sophisticated *developer communities*



Outline

1. Introduction
2. Academia’s evolving practice
3. Connecting communities
4. Challenges
5. Moving forward

Pressure to make the source code available is raising

Why

Necessary to

- *reproduce* and verify,
- *modify* and *evolve*, building new experiments from old ones

Pressure to make the source code available is raising

Why

Necessary to

- *reproduce* and verify,
- *modify* and *evolve*, building new experiments from old ones

When and where

- debate started end of first 2000 decade (biology, statistics, medicine, etc.)
- growing in Computer Science since the ESEC/FSE 2011 Artifact Evaluation context (winner: Vouillon and Di Cosmo)

What is at stake

in increasing order of difficulty

Archival

Research software artifacts must be properly archived

make it sure we can *retrieve* them (*reproducibility*)

What is at stake — in increasing order of difficulty

Archival

Research software artifacts must be properly archived

make it sure we can *retrieve* them (*reproducibility*)

Identification

Research software artifacts must be properly referenced

make it sure we can *identify* them (*reproducibility*)

What is at stake in increasing order of difficulty

Archival

Research software artifacts must be properly archived

make it sure we can *retrieve* them (*reproducibility*)

Identification

Research software artifacts must be properly referenced

make it sure we can *identify* them (*reproducibility*)

Metadata

Research software artifacts must be properly described

make it easy to *discover* them (*visibility*)

What is at stake in increasing order of difficulty

Archival

Research software artifacts must be properly archived

make it sure we can *retrieve* them (*reproducibility*)

Identification

Research software artifacts must be properly referenced

make it sure we can *identify* them (*reproducibility*)

Metadata

Research software artifacts must be properly described

make it easy to *discover* them (*visibility*)

Citation

Research software artifacts must be properly cited *(not the same as referenced!)*

to give *credit* to authors (*evaluation*!)

Where we stand

Lack of recognition

not (yet) a first class citizen

- in the EOSC plan
- in the scholarly world

Where we stand

Lack of recognition

not (yet) a first class citizen

- in the EOSC plan
- in the scholarly world

Lack of consensus on how to

- *archive* software
- *choose* a license
- *cite* a software project

Where we stand

Lack of recognition

not (yet) a first class citizen

- in the EOSC plan
- in the scholarly world

Lack of consensus on how to

- *archive* software
- *choose* a license
- *cite* a software project

… but a wealth of initiatives!

- Policies: ACM Artifact Review and Badging, AEC, …
- Working groups: FORCE11, RDA, SPSO, …
- Journals: IPOL, ReScience, InsightJournal, JOSS, eLife, ACM DL, …
- Repositories: FigShare, Zenodo, …
- Common infrastructures: Software Heritage

Outline

1 Introduction

2 Academia's evolving practice

3 Connecting communities

4 Challenges

5 Moving forward

FORCE Software Citation Implementation WG

Spawned from the Software Citation WG (2/2016)

led by Daniel Katz, Kyle Niemeyer and Arfon Smith

Co-chairs

Neil Chue Hong, Martin Fenner, Daniel Katz

FORCE Software Citation Implementation WG

Spawned from the Software Citation WG (2/2016)

led by Daniel Katz, Kyle Niemeyer and Arfon Smith

Co-chairs

Neil Chue Hong, Martin Fenner, Daniel Katz

. . .

Neil tells us more. . .

RDA Software Source Code Interest Group

Co-chairs

Roberto Di Cosmo, Neil Chue Hong, Mingfang Wu, Julia Collins

Objectives

a forum for discussing *software* inside RDA

Chronology

RDA 10, Montreal 9/2017 motivations, survey of ontologies, metadata use cases

RDA 11, Berlin 3/2018 identification of gaps in metadata

RDA 13, Philadelphia 4/2019 FAIR for Software Source Code

Web page

https://www.rd-alliance.org/groups/software-source-code-ig

RDA WG on Software Source Code Identification

Joint RDA & FORCE11 WG which spawned from

RDA's Software Source Code IG & FORCE11's SCIWG

Co-chairs

Roberto Di Cosmo, Daniel Katz, Martin Fenner

Objectives

- bring together people involved/interested in *software identification*
- produce concrete recommendations for the academic community

`https://www.rd-alliance.org/groups/software-source-code-identification-wg`

Inria's Software Citation Working Group

Members

task force of Inria's scientific council

Mission

- map the landscape
- collect best practices
- identify potential Inria contributions
- make recommendations

First outcome

Position paper available from
https://hal.archives-ouvertes.fr/hal-02135891

Outline

1. Introduction
2. Academia's evolving practice
3. Connecting communities
4. **Challenges**
5. Moving forward

Much more complex than it seems

Software is complex

Structure	monolithic/composite; self-contained/external dependencies
Lifetime	one-shot/long term
Community	one man/one team/distributed community
Authorship	complex set of roles
Authority	institutions/organizations/communities/single person

Much more complex than it seems

Software is complex

Structure monolithic/composite; self-contained/external dependencies

Lifetime one-shot/long term

Community one man/one team/distributed community

Authorship complex set of roles

Authority institutions/organizations/communities/single person

Various granularities

Exact status of the source code for reproducibility, e.g.

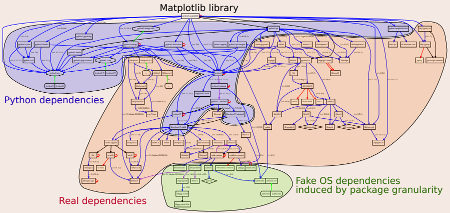
“you can find at swh:1:cnt:cdf19c4487c43c76f3612557d4dc61f9131790a4;lines=146-187 the core algorithm used in this article”

(Major) release *“This functionality is available in OCaml version 4”*

Project *“Inria has created OCaml and Scikit-Learn”.*

We are not alone

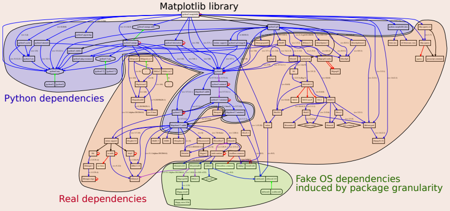
Research Software does not exist in isolation



large *web of dependencies* on non-research software

We are not alone

Research Software does not exist in isolation



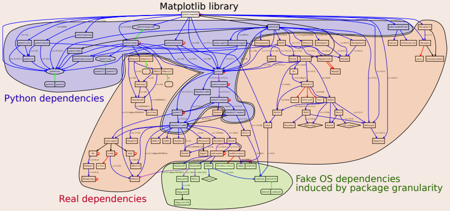
large *web of dependencies* on non-research software

Industry and developers have been here

- NSRL (NIST)
- SPDX (Linux Foundation)
- SWH-ID (Software Heritage)
- SWID (ISO Standard)
- Wikidata Software Properties

We are not alone

Research Software does not exist in isolation



large *web of dependencies* on non-research software

Industry and developers have been here

- NSRL (NIST)
- SPDX (Linux Foundation)
- SWH-ID (Software Heritage)
- SWID (ISO Standard)
- Wikidata Software Properties

We must

- accept the complexity
- avoid reinventing the wheel
- connect with existing communities of practice

Outline

1. Introduction
2. Academia's evolving practice
3. Connecting communities
4. Challenges
5. Moving forward

Our goals today

Make progress

- Share and collect knowledge
- Improve state of the art
- Other tangible outputs, as detailed in the agenda

Thanks, and good work!