Software Source Code Interest Group

Introduction

Roberto Di Cosmo (INRIA), Neil Chue Hong (SSI)

roberto@dicosmo.org

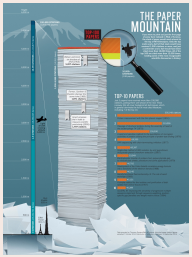
September 19th, 2017

Software Heritage

THE GREAT LIBRARY OF SOURCE CODE

Why we are here

Software is *an essential component* of modern scientific research



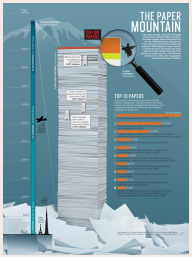
Top 100 papers (Nature, October 2014)

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

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

Why we are here

Software is *an essential component* of modern scientific research



Top 100 papers (Nature, October 2014)

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

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

The *source code* is essential

- it contains the *real knowledge*,
- it is currently poorly accounted for

Reminder: the *source code* of a software artefact

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

GPL Licence

Reminder: the *source code* of a software artefact

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

```
Hello World
```

Reminder: the *source code* of a software artefact

"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
```

Reminder: the *source code* of a software artefact

"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");
}
```

R1: Software Source Code is *special*

Harold Abelson, Structure and Interpretation of Computer Programs

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

Quake 2 source code (excerpt)

```
float Q_rsqrt( float number )
{
    long i;
    float x2, y;
    const float threehalfs = 1.5F;

    x2 = number * 0.5F;
    y = number;
    i = * ( long * ) &y; // evil floating point bit level hacking
    i = 0x5f3759df - ( i >> 1 ); // what the fuck?
    y = * ( float * ) &i;
    y = y * ( threehalfs - ( x2 * y * y ) ); // 1st iteration
// y = y * ( threehalfs - ( x2 * y * y ) ); // 2nd iteration, this
can be removed

    return y;
}
```

Net. queue in Linux (excerpt)

```
/*
 * SFB uses two B[l][n] : L x N arrays of bins (L levels, N bins per level)
 * This implementation uses L = 8 and N = 16
 * This permits us to split one 32bit hash (provided per packet by rxhash or
 * external classifier) into 8 subhashes of 4 bits.
 */
#define SFB_BUCKET_SHIFT 4
#define SFB_NUMBUCKETS  (1 << SFB_BUCKET_SHIFT) /* N bins per Level */
#define SFB_BUCKET_MASK (SFB_NUMBUCKETS - 1)
#define SFB_LEVELS      (32 / SFB_BUCKET_SHIFT) /* L */

/* SFB algo uses a virtual queue, named "bin" */
struct sfb_bucket {
        u16             qlen; /* length of virtual queue */
        u16             p_mark; /* marking probability */
};
```

Len Shustek, Computer History Museum

"Source code provides a view into the mind of the designer."

R2: Source code is not … just data

R2: Source code is not … just data

executable and *human readable* knowledge (an *all time new*)

- written *by humans for humans*
- formats not really an issue: *text files are forever*

R2: Source code is not … just data

executable and *human readable* knowledge (an *all time new*)

- written *by humans for humans*
- formats not really an issue: *text files are forever*

the *development history* is key to its *understanding*

- version history
- literate programming

R2: Source code is not … just data

executable and *human readable* knowledge (an *all time new*)

- written *by humans for humans*
- formats not really an issue: *text files are forever*

the *development history* is key to its *understanding*

- version history
- literate programming

complexity:

- large *web of dependencies*
- millions of SLOCs

R2: Source code is not … just data

executable and _human readable_ knowledge (an _all time new_)

- written _by humans for humans_
- formats not really an issue: _text files are forever_

the _development history_ is key to its _understanding_

- version history
- literate programming

complexity:

- large _web of dependencies_
- millions of SLOCs

Bottomline: software source code _is not just another_ sequence of bits

R3: we are not taking care of it

No universal catalog

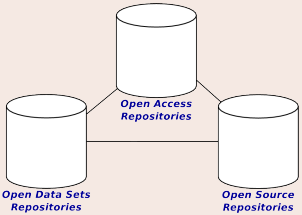
No universal archive

R3: we are not taking care of it

No universal catalog

No universal archive

The Knowledge Conservancy Magic Triangle



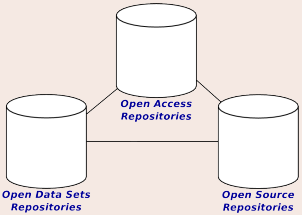
- Articles: HAL, ArXiv, 100s of inst. repositories
- Data: Zenodo, Figshare, 100s of various repositories
- Software:

R3: we are not taking care of it

No universal catalog

No universal archive

The Knowledge Conservancy Magic Triangle



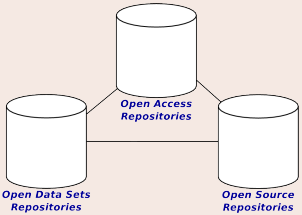
- Articles: HAL, ArXiv, 100s of inst. repositories
- Data: Zenodo, Figshare, 100s of various repositories
- Software:
  - R4: GitHub does not fit the bill

R3: we are not taking care of it

No universal catalog

No universal archive

The Knowledge Conservancy Magic Triangle



- Articles: HAL, ArXiv, 100s of inst. repositories
- Data: Zenodo, Figshare, 100s of various repositories
- Software:
  - R4: GitHub does not fit the bill
  - R5: we want to avoid duplication of efforts

RDA is a good place for starting the conversation on...

Metadata

- what kind of *ontology* exist for software?
- what would be appropriate for Source Code?

RDA is a good place for starting the conversation on...

Metadata

- what kind of *ontology* exist for software?
- what would be appropriate for Source Code?

Use cases

- discovery
- citation
- classification
- documentation, ...

RDA is a good place for starting the conversation on…

Metadata

- what kind of *ontology* exist for software?
- what would be appropriate for Source Code?

Use cases

- discovery
- citation
- classification
- documentation, …

Relation to professional software development

- is scientific software different from, say, usual open source software?
- can we learn from the experience of millions of open source developers?

Objectives and Agenda

Objectives

- metadata frameworks for source code
  - analyze and identify gaps
- collect use cases

Agenda

1. Introduction (done)
2. Overview of metadata frameworks for source code
3. Parallel discussion and gap identification
4. Collection of potential use cases
5. Summary of results and wrap up

Reminder

RDA SCIG page

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

Working document used during the session

https://bit.ly/RDA10SoftwareIGNotes