# Curated Archiving of Research Software Artifacts:

lessons learned from the French open archive (HAL)

Roberto Di Cosmo, **Morane Gruenpeter**, Bruno Marmol,
Alain Monteil, Laurent Romary, Jozefina Sadowska

February 18th, 2020

## Outline

## Source Code: *executable* and *human readable* knowledge

*"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");
}
```

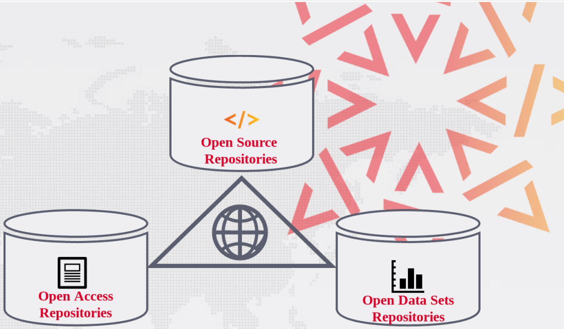
## Software is a *forgotten* pillar of Open Science

### Lack of recognition

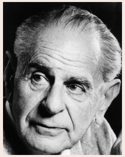
not (yet) a first class citizen

- in the EOSC plan
- in the scholarly world

*Sometimes, if you don't have the software, you don't have the data*
*Christine Borgman, Paris, 2018*



### Reproducibility is the key



*non-reproducible single occurrences are of no significance to science*

*Karl Popper, The Logic of Scientific Discovery, 1934*

## 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*!)

## Outline

# Making software a first class research output

## CCSD

Center for Direct Scientific Communication - behind the HAL platform

- Hyper articles en ligne

## IES-Inria

Scientific information & publishing service @Inria

- Institut National de Recherche en Informatique et en Automatique

## Software Heritage

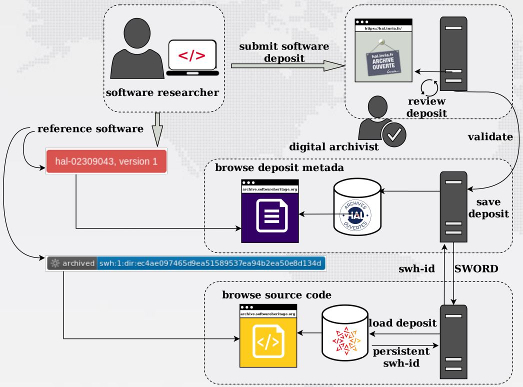
Building the SWH universal archive for all *software source code*

- With the support of UNESCO

## Goals

1. archive software source code on HAL and on SWH
2. identify all the contained artifacts in a deposit with the *SWH-ID*
3. describe with reviewed metadata by an *IES-Inria moderator*
4. cite the deposit with a complete citation

## The research software (deposit) use case



### Deposit software in HAL *poster*

Generic mechanism:

- SWORD based
- review process
- versioning

How to do it: (*guide*)

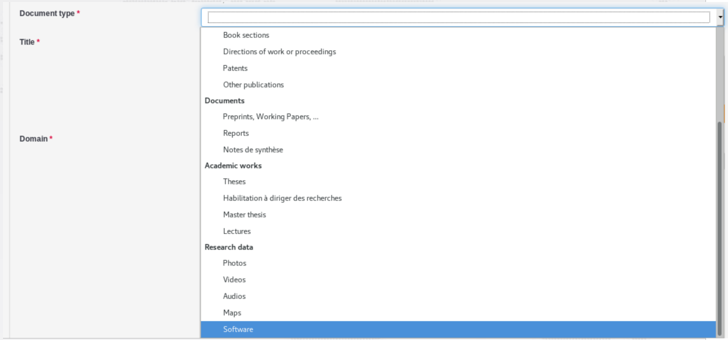
- deposit .zip or .tar.gz file with metadata

Timeline:

- *March 2018*: test phase on HAL-Inria
- *September 2018*: open to all HAL
- *December 2019*:
  - 80 complete source code deposits
  - 98 software records

# Submit your source code

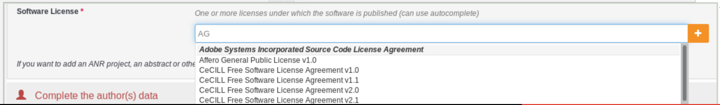
(*guide*)


Document type *

Title *

Domain *

Book sections
Directions of work or proceedings
Patents
Other publications
**Documents**
Preprints, Working Papers, ...
Reports
Notes de synthèse
**Academic works**
Theses
Habilitation à diriger des recherches
Master thesis
Lectures
**Research data**
Photos
Videos
Audios
Maps
Software


Software License *

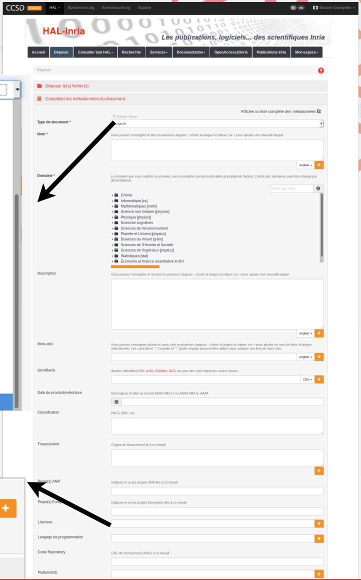
*One or more licenses under which the software is published (can use autocomplete)*

AG

***Adobe Systems Incorporated Source Code License Agreement***
Affero General Public License v1.0
CeCILL Free Software License Agreement v1.0
CeCILL Free Software License Agreement v1.1
CeCILL Free Software License Agreement v2.0
CeCILL Free Software License Agreement v2.1

*If you want to add an ANR project, an abstract or othe*

Complete the author(s) data



Déposer

Déposer le(s) fichier(s)

Compléter les métadonnées du document

Afficher la liste complète des métadonnées

* Champs requis

Type de document * Logiciel

Nom * *Vous pouvez renseigner le titre en plusieurs langues : choisir la langue et cliquer sur + pour ajouter une nouvelle langue*

anglais

Domaine * *Le domaine que vous mettrez en premier, sera considéré comme la discipline principale de l'article. L'ordre des domaines peut être changé par glissé/déposé.*

Filtrer par nom

Chimie
Informatique [cs]
Mathématiques [math]
Science non linéaire [physics]
Physique [physics]
Sciences cognitives
Sciences de l'environnement
Planète et Univers [physics]
Sciences du Vivant [q-bio]
Sciences de l'Homme et Société
Sciences de l'ingénieur [physics]
Statistiques [stat]
Économie et finance quantitative [q-fin]

Description *Vous pouvez renseigner le résumé en plusieurs langues : choisir la langue et cliquer sur + pour ajouter une nouvelle langue*

anglais

Mots-clés *Vous pouvez renseigner plusieurs mots-clés en plusieurs langues : choisir la langue et cliquer sur + pour ajouter un mot-clé dans la langue sélectionnée. Les caractères "," (virgule) et ";" (point-virgule) peuvent être utilisés pour séparer une liste de mots-clés.*

anglais

Identifiants *Ajoutez l'identifiant DOI, arXiv, PubMed, ADS, etc pour lier votre dépôt aux autres bases.*

DOI

Date de production/écriture *Renseignez la date au format AAAA-MM-JJ ou AAAA-MM ou AAAA*

Classification *PACS, MSC, etc.*

Financement *Origine du financement lié à ce travail*

Projet(s) ANR *Indiquez le ou les projets ANR liés à ce travail*

Projet(s) Euro *Indiquez le ou les projets Européens liés à ce travail*

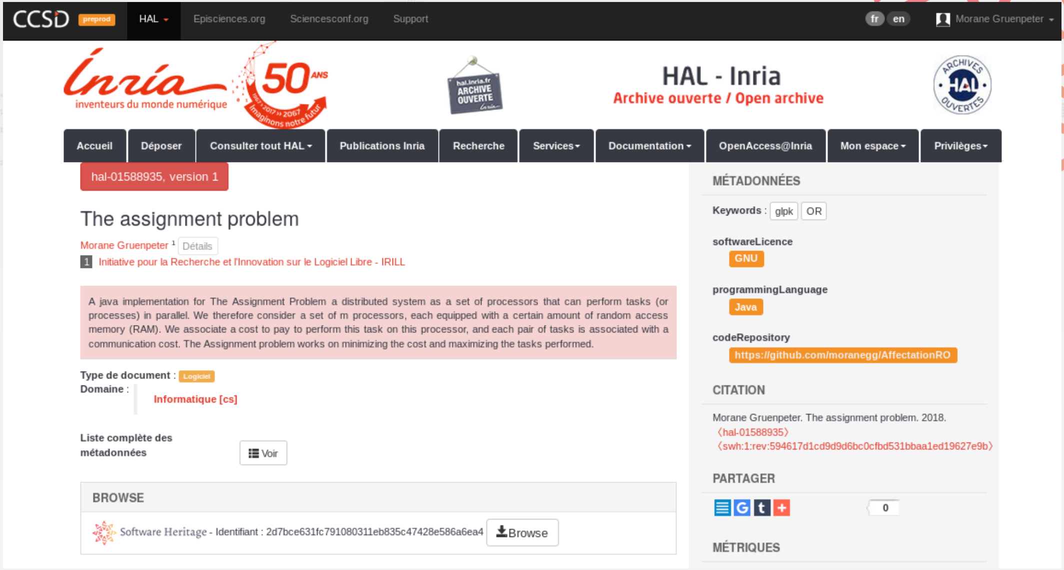
Licences

Langage de programmation

Code Repository *URL de ressource(s) liée(s) à ce travail*

Platform/OS

# The deposit view



hal-01588935, version 1

## The assignment problem

Morane Gruenpeter [1] Détails

[1] Initiative pour la Recherche et l'Innovation sur le Logiciel Libre - IRILL

A java implementation for The Assignment Problem a distributed system as a set of processors that can perform tasks (or processes) in parallel. We therefore consider a set of m processors, each equipped with a certain amount of random access memory (RAM). We associate a cost to pay to perform this task on this processor, and each pair of tasks is associated with a communication cost. The Assignment problem works on minimizing the cost and maximizing the tasks performed.

Type de document : Logiciel

Domaine : Informatique [cs]

Liste complète des métadonnées: Voir

BROWSE

Software Heritage - Identifiant : 2d7bce631fc791080311eb835c47428e586a6ea4 Browse

MÉTADONNÉES

Keywords : glpk OR

softwareLicence: GNU

programmingLanguage: Java

codeRepository: https://github.com/moranegg/AffectationRO

CITATION

Morane Gruenpeter. The assignment problem. 2018. 〈hal-01588935〉 〈swh:1:rev:594617d1cd9d9d6bc0cfbd531bbaa1ed19627e9b〉

PARTAGER

0

MÉTRIQUES

## Reference vs. citation

**Credit & Attribution**

- a metadata record
- all authors & contributors

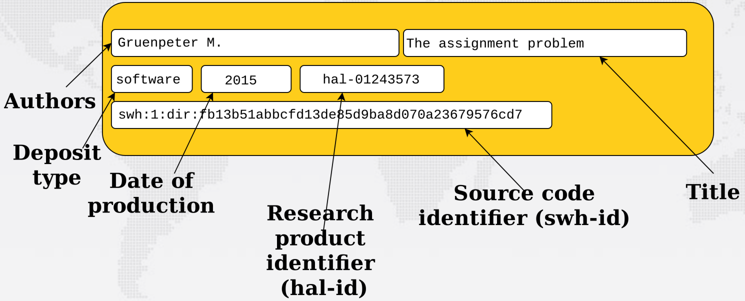
**Reuse & Reproducibility**

- a specific artifact
- with complementary information (docs)

**Archive & Index**

- metadata record (HAL)
- artifact itself (SWH)

connect the dots...


Gruenpeter M.
The assignment problem
software
2015
hal-01243573
swh:1:dir:fb13b51abbcfd13de85d9ba8d070a23679576cd7
Authors
Deposit type
Date of production
Research product identifier (hal-id)
Source code identifier (swh-id)
Title

## Outline

## Software deposit moderation

### we need

- quality metadata to describe research software
- correct credit to all authors of the software

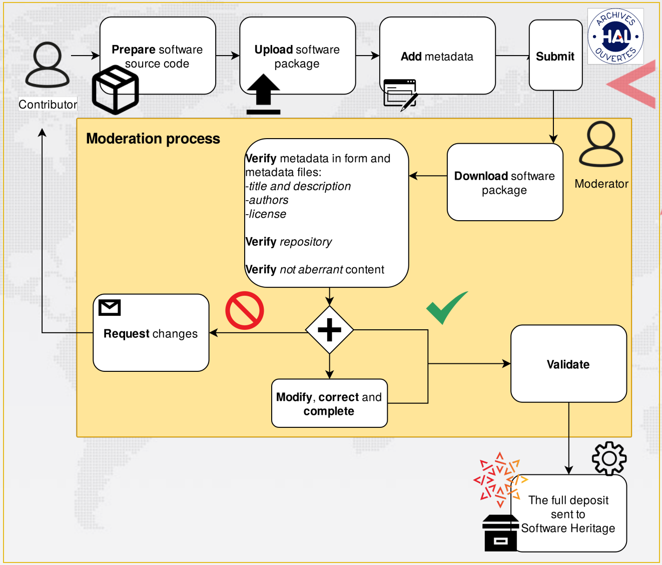
### Main actions the digital archivist performs:

- detecting extraneous or abusive content (illegal or harassing),
- verifying consistency between the metadata and the software source code itself,
- completing or correcting the deposit metadata if needed.

### Out of scope

- review source code functionality
- compile & run software
- assess reproducibility & accuracy

## The people behind the scenes


Contributor
**Prepare** software source code
**Upload** software package
**Add** metadata
**Submit**
ARCHIVES HAL OUVERTES
**Moderation process**
Moderator
**Download** software package
**Verify** metadata in form and metadata files:
-*title and description*
-*authors*
-*license*
**Verify** *repository*
**Verify** *not aberrant* content
**Request** changes
**Modify**, **correct** and **complete**
**Validate**
The full deposit sent to Software Heritage

## Publishing vs Sharing

### Publishing

- a research result that has been qualified through peer review
- software review examples:
  - AEC,
  - IPOL Journal · Image Processing On Line IPOL,
  - JOSS- the Journal of Open Source Software
- still new in the scholarly ecosystem for software

### Sharing

- vast majority developed outside of academia
- on fashionable code hosting platforms with no long-term guarantees (Github, Gitlab, Gitorious. . . )
- research software preservation with institutional repositories or archives (HAL, Zenodo, SWH, etc..)
- don't require review process

Sharing is crucial for Open Science

## Outline

## Lessons learned

### The importance of a software license

- can software be deposited without a license?

became a mandatory field on HAL

### Collective authorship

- can the X project team be the author of software?

authorship can be established only with a clear link between a *person and a deposit*

### Legacy software

- should be archived in its original state
- where to put additional information?

create source code container to capture both *original* and *added information* as detailed in the legacy software acquisition process (SWHAP)

# Lessons learned (continued..)

## research experiments

- deposit on HAL or just archive repository on SWH?

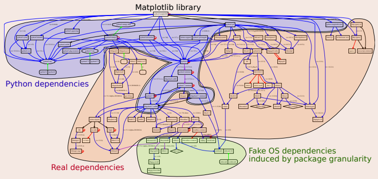
depends on the life span of the experiment

## software with large datasets

- include in software deposit or separate?

depends on dataset nature and reuse possibilities

## Software collections



- Research Software does not exist in isolation
- large *web of dependencies* on non-research software
- single or multiple deposits ?

depends on reuse possibilities

## Next steps

### Export formats

- improve BibTex export (contribute to the @software bibtex proposal)
- improve other existing formats (TEI, endnote, DC, DCterms)
- create CodeMeta and CFF exports of metadata

### Create deposit from existing repository

- using an existing SWH-ID
- using a repository url (on GitHub, GitLab, etc..)-> Save Code Now

### Integrate software into HAL Data

- on `https://data.archives-ouvertes.fr/`
- a SPARQL endpoint, using RDF

## Come in, we're open!

This work is partially supported by the FAIRsFAIR European project.

Questions?

R. Di Cosmo, M. Gruenpeter, S. Zacchiroli
*Referencing Source Code Artifacts: a Separate Concern in Software Citation*,
CiSE, IEEE, pp.1-9. 2020. (10.1109/MCSE.2019.2963148) (hal-02446202)

R. Cosmo, M. Gruenpeter, B. Marmol, A. Monteil, L. Romary, J. Sadowska
*Curated Archiving of Research Software Artifacts: lessons learned from the French open archive*,
submitted to IJDC. December 2019. (hal-02475835)

P. Alliez, R. Di Cosmo, B. Guedj, A. Girault, M.-S. Hacid, A. Legrand, N. Rougier
*Attributing and Referencing (Research) Software: Best Practices and Outlook From Inria Journal Article*,
Computing in Science Engineering, 22 (1), pp. 39-52, 2020, ISSN: 1558-366X.
(10.1109/MCSE.2019.294941) (hal-02135891)