



Lier codes de la recherche et publications via HAL

Journée des correspondants HAL
25 avril 2023

Plan

- **Enjeux**
 - Software Heritage et HAL
- **Signaler un code sur HAL**
- **Lier publications et codes**
- **Bonnes pratiques**

Enjeux du référencement des codes de recherche

Pourquoi référencer les codes de recherche ?

■ Pourquoi archiver ?

- Le code source est fragile :
 - Obsolescence des formats, problème matériel, dépendances à des outils (forge par exemple) qui disparaissent ...
 - La perte des codes ayant été utilisés pour de la production scientifique arrive malheureusement régulièrement
- Les logiciels sont un des piliers des processus de recherche, au côté des publications et des données et il est essentiel de les préserver

■ Pourquoi signaler ?

- Assurer la description
- Faciliter la recherche / Rechercher facilement (par domaine scientifique en particulier)
- Permettre la citation
- Valoriser les logiciels

Software Heritage



- Initiative dont l'objectif est de construire une **archive universelle des codes sources**
- En les collectant, les préservant et les partageant sur le long terme
- Lancée en 2016 par INRIA et soutenue par l'UNESCO
- Collecte de l'**intégralité des logiciels disponibles publiquement** sous forme de code source.
- Depuis des plateformes d'hébergement de code, comme GitHub, GitLab.com ou Bitbucket, et des archives de paquets, comme Npm ou Pypi ...



HAL + Software Heritage

■ Un nouveau type de dépôt sur HAL

- Collaboration initiée en 2018 entre HAL et SWH
- Phase de test avec INRIA
- En particulier, le dépôt SWHID en production depuis le début d'année

■ Complémentarité des deux plateformes

- Grande visibilité des logiciels dans une démarche de science ouverte via HAL
- Archivage pérenne via Software Heritage
- Modération des métadonnées
- Différents formats d'export pour faciliter la citation

Exemple d'un code de recherche référencé



Consulter sur Software Heritage

Logiciel Année :

Dates et versions

hal-04056972, version 1 (03-04-2023)

Identifiants

HAL Id : hal-04056972, version 1
SWHID : swh:1:dir:e4e31bb51de8898376e7dce8206c0e6a7176186e;origin=https://github.com/siconos/siconos;visit=swh:1:snp:d9057646a859f43c1eae31cb4c7d7dd3030a440d;anchor=swh:1:rev:6ca5eb6134478ab9cf0e4a5f259d602e52e3df3a

Métadonnées

- version : 4.4.0
- Licences : <https://spdx.org/licenses/Apache-2.0>
- Langage de programmation : Python, C++, Fortran, C
- Code Repository : <https://github.com/siconos/siconos>

Citer

Siconos

Vincent Acary (1), Maurice Bremond (2), Olivier Huber (3), Franck Perignon (4), Roger Pissard-Gibollet (2)

[Afficher plus de détails](#)

- 1 TRIPOP - Modélisation, simulation et commande des systèmes dynamiques non lisses
- 2 SED [Grenoble] - Service Expérimentation et Développement
- 3 WIAS - Weierstraß-Institut für Angewandte Analysis und Stochastik = Weierstrass Institute for Applied Analysis and Stochastics [Berlin]
- 4 LJK - Laboratoire Jean Kuntzmann

Résumé en

Siconos is an open-source scientific software primarily targeted at modeling and simulating nonsmooth dynamical systems in C++ and in Python

Mots clés en

Nonsmooth systems frictional contact
complementarity

Domaines

Mathématiques [math]

[Liste complète des métadonnées](#)

Historique

<https://github.com/siconos/siconos>

15 April 2023, 20:08:51 UTC

<> Code Branches (41) Releases (4) Visits

Revision: `6ca5eb6134478ab9cf0e4a5f259d602e52e3df3a` e4e31bb / [History](#) [Download](#) [Save again](#)

Tip revision: `6ca5eb6134478ab9cf0e4a5f259d602e52e3df3a` authored by Vincent ACARY on 27 January 2023, 17:05:55 UTC [numerics] typo

File	Mode	Size
.github		
ci_gitlab		
siconos.spec	-rw-r--r--	2.5 KB

README.md

Siconos

A software package for the modeling and simulation of nonsmooth dynamical systems in C++ and in Python.

Siconos is an open-source scientific software primarily targeted at modeling and simulating nonsmooth dynamical systems:

Signaler un code sur HAL

Préalables

- **On s'intéresse ici au dépôt via SWHID**
 - Cela suppose l'utilisation d'une forge (qui soit moissonnée par SWH) : c'est **LA bonne pratique pour le développement des codes**
 - **Processus SWH puis HAL** : Les **informations sur la forge sont toujours à jour** donc l'archive sur SWH l'est aussi (possibilité de forcer la mise à jour) et il reste à signaler sur HAL
 - A noter qu'il est aussi possible de déposer via une archive zip sur HAL. Dans ce cas les fichiers des codes sont automatiquement archivés sur SWH.
- **Que déposer ?**
 - Le code source
 - D'un logiciel développé dans un milieu académique

Checklist : préparer le logiciel

- **Métadonnées obligatoires**
 - Nom du logiciel
 - AUTHORS
 - LICENSE
 - Fichier README

- Existence d'un identifiant unique : le **SWHID**

Signalement sur HAL

■ Nouveau dépôt → Logiciel

- Ne pas déposer de fichier mais indiquer le **SWHID avec contexte** (pour garder le lien vers l'historique de développement)

</> **Logiciel** [changer](#)

Métadonnées, Je renseigne mon dépôt

[Extraction automatique](#)

Chargez les métadonnées à partir d'un identifiant

Les informations associées à cet identifiant permettront de compléter automatiquement votre dépôt.

SWHID ▾

Récupérer les métadonnées

Signalement sur HAL

- **Vérifier les métadonnées insérées** automatiquement et **compléter les métadonnées** (par ex, affiliations, financements, domaines, etc.)
- Vérifier les **auteurs et les affiliations**
- **Faire les liens avec les publications associés aux codes et avec les données**
- **Valider** le dépôt

Lier codes et publications

Liens codes, publications, et codes

- Indication des publications dans le README sur **Software Heritage**

Identifiants

Identifiants

Ajoutez l'identifiant [DOI](#), [arXiv](#), [PubMed](#), [ADS](#), etc pour lier votre dépôt aux autres bases.

SWHID ▾



Données associées

Ajoutez l'identifiant [DOI](#) fourni par l'entrepôt où vos données sont archivées.

← HAL

Métadonnées de workflow de calcul ^

Type de workflow ?

URL vers le dépôt de code externe ?

Documentation ?

Recherche Data Gouv →

Bonnes pratiques

Bonnes pratiques

■ Pour les publications

- déposer le texte intégral (version validée pour publication)
- Veiller à indiquer le doi

■ Pour les données

- Les déposer dans Recherche Data Gouv / autres entrepôts
- Indiquer systématiquement pour les jeux de données le doi et l'identifiant HAL de la publication associée
- Indiquer sur la publication déposée dans HAL le doi correspondant au jeu de données déposé

■ Pour les codes :

- Veiller à leur archivage dans SWH
- Les signaler dans HAL
- Indiquer les doi et/ou identifiants HAL des publications et des jeux de données associés

En résumé : pour vous aider

■ Des adresses supports

- Pour les publications : **hal.support@univ-grenoble-alpes.fr**
- Pour les données : **sos-data@univ-grenoble-alpes.fr**
- Pour les codes : **sos-codes-recherche@univ-grenoble-alpes.fr**

Ressources

Références

- Create software deposit in HAL: User guide and best practices , <https://hal.archives-ouvertes.fr/hal-01872189>
- Modérer un dépôt logiciel dans HAL : dépôt source et dépôt SWHID, <https://hal.inria.fr/hal-01876705v2>
- HOWTO archive and reference your code, <https://www.softwareheritage.org/howto-archive-and-reference-your-code/>
- Depositing scientific software into Software Heritage, <https://www.softwareheritage.org/2018/09/28/depositing-scientific-software-into-software-heritage/>
- Tutoriaux vidéos : <https://youtube.com/playlist?list=PLD2VqrZz2-u3bOWtoCoBIh5Flt6iYXsq3>
- Save code now : <https://archive.softwareheritage.org/save/>
- Merci à Morane Gruenpeter et à Sabrina Granger de Software Heritage !