Fingerprinting and Building Large Reproducible Datasets

ACM REP '23

Romain Lefeuvre
University of Rennes
France
romain.lefeuvre@inria.fr

Jessie Galasso
DIRO, Université de Montréal
Canada
jessie.galasso-carbonnel@umontreal.ca

Benoit Combemale
University of Rennes
France
benoit.combemale@irisa.fr

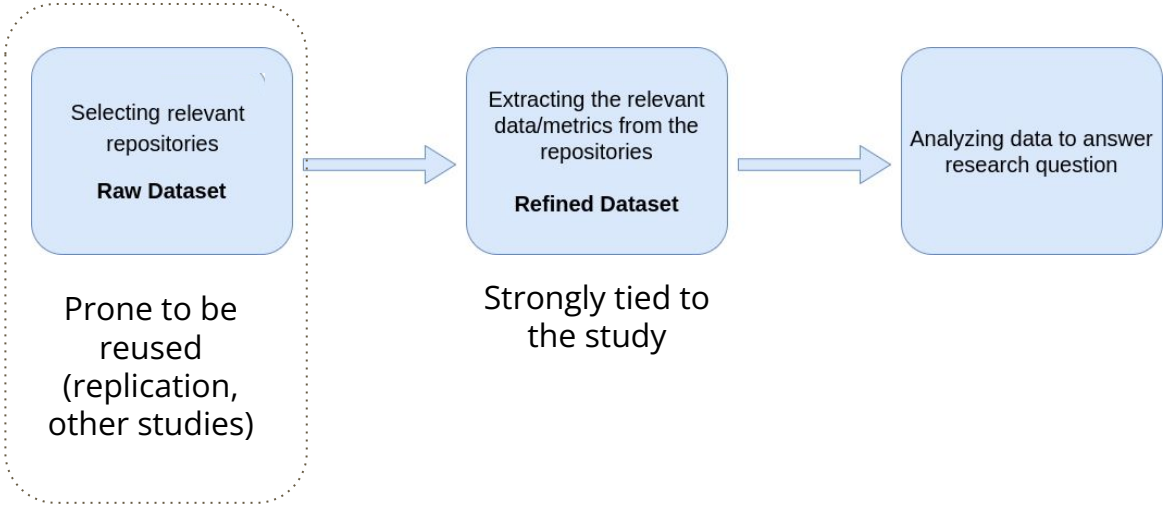
Houari Sahraoui
DIRO, Université de Montréal
Canada
sahraouh@iro.umontreal.ca

Stefano Zacchiroli
LTCI, Télécom Paris, Institut Polytechnique de Paris
France
stefano.zacchiroli@telecom-paris.fr

Reproducibility & empirical research in SE

- Growing interest for empirical research in Software Engineering
- Obtaining a relevant dataset is key
- All the major conferences consider reproducibility as an evaluation factor

Big picture of a typical empirical study in SE



Focus on dataset composed of source code from public repositories

Limitations when reusing raw datasets

Impossible to ensure reproducibility of datasets that include links to resources changing over time

- **Source code evolves over time**

  → Provide timestamp/hash to retrieve the state of the repository?

- **Projects can be deleted or the history can be rewritten**

  → Hash is not enough , snapshot ?

Limitations when **reproducing** an existing raw dataset

Sometimes it's necessary to reproduce the steps for selecting the repositories but it is often a complex process since:

- The **selection process is not systematic** and/or not clearly defined
- The data sources are **not reliable** and do not ensure reproducibility
- The API provided by traditional forges are **not adapted for large scale empirical studies**

Github Search API is not a reliable source of information

- The same query executed twice **3 Millions to 9 Millions results**
- Search API restriction (query must return less than 1k results ...)

GitHub Code Search Now Generally Available, 'Way More than grep'

By David Ramel 05/09/2023

- **A new code search engine** rebuilt completely from scratch,

Limitations when creating new Datasets

Forges do not provide appropriate tooling for large scale mining

Heterogeneous information sources with heterogeneous API

SOURCEFORGE

GitLab

Bitbucket

WIKIMEDIA FOUNDATION

... At the end you will choose github

- Query Expressivity Limitation
- Rate Limitation
- Complex API

Software Heritage : Towards the universal software archive

Collect

Sources files

15,779,766,829

Share

Commits

3,278,537,726

Preserve

Projects

241,364,430

Why to use Software Heritage for reproducibility ?

Availability

Multistakeholder infrastructure

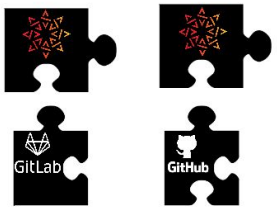
Traceability

Intrinsic unique identifiers (SWHID)

Immutability

Append only model (except law requirement)

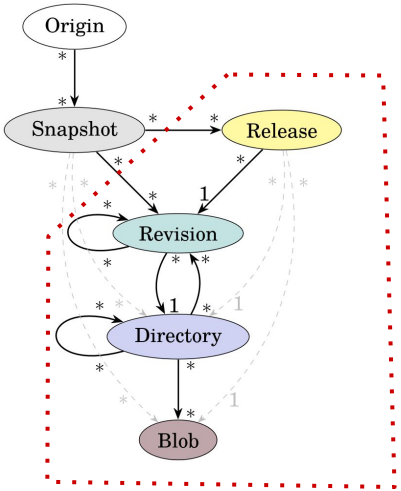
Uniformity



Uniform API

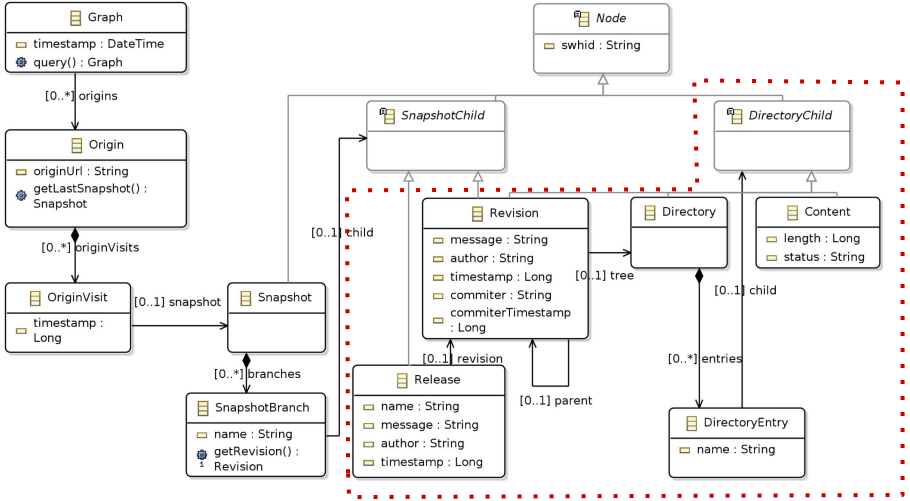
How to use the Software Heritage Graph Dataset

An Open API allowing to query locally the entirety of the model



Software Heritage Graph Dataset [1]

Git structure



Object model of the Software Heritage Graph Dataset

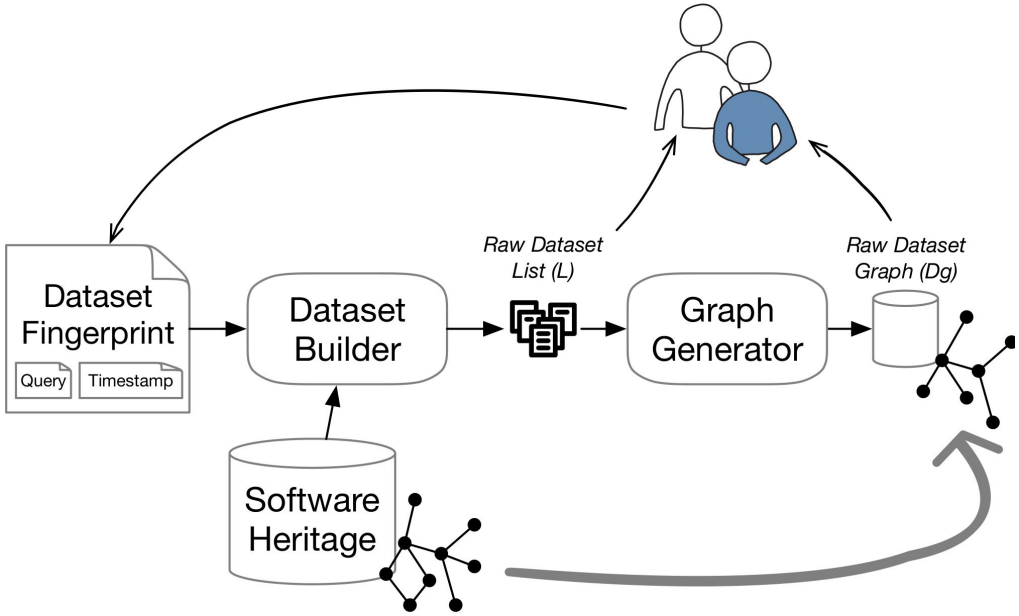
[1] Antoine Pietri. Organizing the graph of public software development for large-scale mining. Université Paris Cité, 2021.

The fingerprint approach

1) A query on the data model of the source code

2) A timestamp to freeze the state of the archive

3) A hash to prevent any corruption



Operationalizing our approach

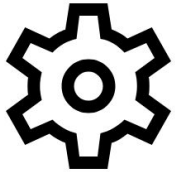
Fingerprint Query Specification

Object Constraint Language (OCL)

Fingerprint Timestamp

SWH Graph Dataset Timestamp

Fingerprint Engine

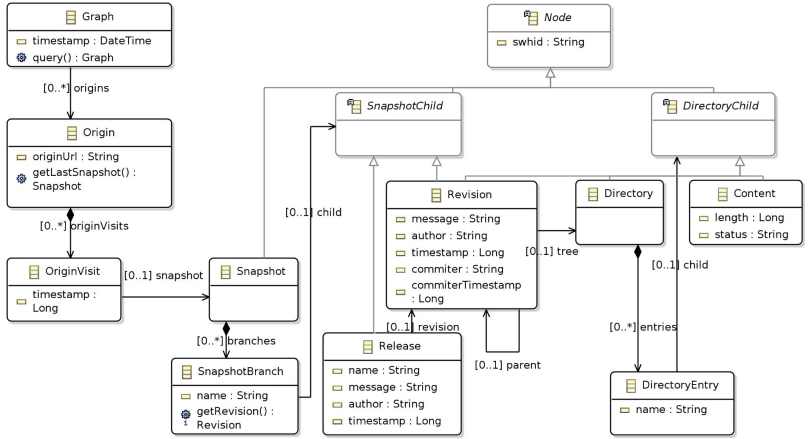


Query compiler to SWH-Graph java API

Operationalization of our approach :

Fingerprint Query Specification

Object Constraint Language (OCL)



Object model of the SWH Graph Dataset

Fingerprint query = constraint on the SWH Graph Dataset

Operationalization of our approach:

Fingerprint Timestamp

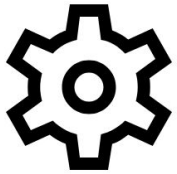
SWH Graph Dataset - Export Timestamp

- Reproducibility ensured on the same export of the SWH Graph Dataset
- Theoretically possible to reconstruct previous states of the SWH Graph Dataset from a more recent export

Fingerprint Timestamp = Frozen state of the SWH Graph Dataset

Operationalization of our approach:

Fingerprint Engine



Query compiler to SWH-Graph java API

- Compile OCL constraint on SWH-Graph to the Object-Oriented wrapper
- Return the list of repositories matched by the query
- Open the way to extract the returned sub graph

Validation

- RQ1: What is the impact of the temporal dimension of the fingerprint on the extracted dataset?
- RQ2: Is the implemented selection process deterministic?
- RQ3: Is it possible to retrieve the same dataset when applying the fingerprint on different versions of the SWH archive?

RQ1 : Impact of the temporal dimension

Variation on the temporal dimension has a huge impact on the number of results

Forge	FP1(2018)	FP2(2021)	FP3(2022)
github.com	830	135820	172012
gitlab.com	3	67	1154
bitbucket.org	-	76	106
codeberg.org	-	55	84
framagit.org	-	21	23
git.launchpad.net	-	10	14
anongit.kde.org	-	2	3
gitlab.gnome.org	-	1	3
git.zx2c4.com	-	1	2
repo.or.cz	-	1	1
gitlab.freedesktop.org	-	-	14
0xacab.org	-	-	3
git.code.sf.net	-	-	3
Total	**833**	**136054**	**173422**

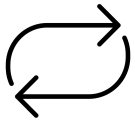
Number of repositories found when executing the same query with different timestamps (FP1=2018, FP2=2021, FP3=2022)

x200 between 2018 and 2022

+27 % between 2021 and 2022

increase in the number of supported forges & forge coverage

RQ2: Determinism of the approach



Executing the same fingerprint over the same export returns the same result

RQ3: Reproducing a dataset over time

Forge	FP3 X G3	FP3 X G4	Difference (%)
github.com	172012	166630	-3.2
gitlab.com	1154	1223	5.6
bitbucket.org	106	102	-3.9
codeberg.org	84	84	0.0
framagit.org	23	22	-4.5
...	43	38	-13.2
Total	**173422**	**168099**	**-3.2**

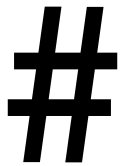
Number of Repositories found when executing the same fingerprint over different versions of the graph

Almost possible to run a fingerprint having an older timestamp than the used graph

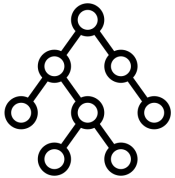
Conclusion

- Approach to **address reproducibility** concerns when creating / reusing / reproducing raw dataset
- Fingerprint **characterizing a dataset** and ensuring to extract the same dataset over time
- Implemented **prototype** leveraging on SWH and OCL

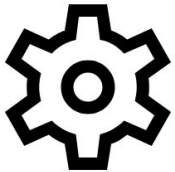
Perspectives



Hash integrity verification



Cover more OCL concepts



Create metrics/index in addition to filter

Thank you for your attention !

Image Credit

Flaticon.com