From Code Repositories to Knowledge Graphs of Research Software Metadata

Daniel Garijo, Ontology Engineering Group, Universidad Politécnica de Madrid, Spain

daniel.garijo@upm.es
@dgarijov
Research Software is one of the pillars of Open Science.

Scientific Publication

Research Data

Research Software

Research Methods
Research Software is one of the pillars of Open Science.
Research Software is one of the pillars of Open Science

Scientific publication

Research Data

Research Software

Research Methods

Research Software: GitHub, Apache, GitLab, Bitbucket, WorkflowHub

Research Data: Figshare, DRYAD, Dataverse Network, Zenodo
Is this situation familiar?

Don't worry, you don't have to start your code from scratch.

You can re-use the software that the previous person on the project wrote several years ago.

Are there instructions for how to use it?

I doubt it. Is the code commented?

Not likely. Where are the files?

Who knows.

This is going to be painful, isn't it?

Just a scratch.

WWW.PHDCOMICS.COM
Challenge 1: Lack of structured (machine-readable) documentation

Don't worry, you don't have to start your code from scratch.

You can re-use the software that the previous person on the project wrote several years ago.

Are there instructions for how to use it?

No doubt it.

Is the code commented?

Not likely.

Where are the files?

Who knows.

GitHub, Gitlab, Docker image? etc.

Readme file, paper(s)

Months
In [1] we tried to reproduce an effort from one year before.

- All data were available online
- All tools were available online (except one, but authors had a replacement)
- > 250 hrs to full reproducibility
- > 100 hrs to get familiar with the tools and their I/O

Challenge 3: Comparing against existing tools

Millions of open-source repositories are updated/created every year

Number of first-time contributors for open source projects, by year

Source: GitHub Octoverse Report 2021
[Hucka et al]: Scientists still rely on three main methods for searching new software:

- Survey
- Recommendation from a colleague
- Search engine

1. Structured representation
2. Reuse
3. Compare
4. Search
1. Structured representation
2. Reuse
3. Compare
4. Search

Knowledge Graphs
Creating KGs of Research Software metadata: Representing RS at different levels of abstraction
Basic Software Metadata

Why is it needed?
- Search
- Compare

creator / contributor
license
description/title/ abstract
keywords
creation date / modified date
publisher / organization
...

ORCIDs not available

https://w3id.org/okn/o/sd
https://w3id.org/software-types/
Software and versions

**Code Repository**

- hasVersion
  - previous
    - publishedIn
      - package indices
  - previous
    - publishedIn
  - publishedIn
  - latest
    - deposited In
      - Zenodo

**Why is it needed?**
- Search
- Compare
- Reuse

**Which identifiers to use?**
Why is it needed?
- Credit
- Trust

How to reconcile code authors with paper authors?
Provenance and code contributions

Why is it needed?
- Credit
- Responsibility
- Accountability

Contributions?
Development process?
Describing inputs, outputs and their structure

Why is it needed?
- Reuse
- Reproducibility

Adapt new sources

Visualize result

Penn State Hydrology Model (PIHM)

Weather • DEM • Infiltration

Outflow • Error

FLDAS (climate) • Remote sensing
Dependencies

Why is it needed?
- Reuse
- Security

```python
install_requires = [
    "bs4==0.8.1",
    "Click==7.0",
    "click-option-group==0.5.3",
    "markdown==3.3.6",
    "matplotlib==3.5.0",
    "nltk==3.6.6",
    "numpy==1.22.0",
    "pandas==1.3.4",
    "rdflib==6.0.2",
    "rdflib-jsonld==0.6.2",
    "requests>=2.22.0",
    "scikit-learn==1.0",
    "textblob==0.17.1",
    "validators==0.18.2",
    "xgboost==1.5.0"
]
```
Software images are created from configuration files (e.g., Dockerfiles)

Fig. by Jhon Toledo

Initial effort transforming part of DockerHub: [https://dockerpedia.inf.utfsm.cl/](https://dockerpedia.inf.utfsm.cl/)

Why is it needed?
- Reuse
- Security

Creating KGs of Research Software metadata:
Knowledge extraction
Research Software metadata is not abundant and machine readable

Can you please describe your software component with metadata?

I already did! Did you read the project readme?

Did you see the online documentation?

Perhaps the you saw the paper?

Many domain-specific registries are curated by hand by experts
### Extracting knowledge from software projects

- **Documentation**
  - Text classification
  - Named entity recognition and relation extraction

- **Code**
  - Static code analysis

<table>
<thead>
<tr>
<th>Directory</th>
<th>File</th>
<th>Action</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>docs</td>
<td></td>
<td>update doc</td>
<td>13 days ago</td>
</tr>
<tr>
<td>experiments</td>
<td></td>
<td>Added pipeline missed in previous version of create_models</td>
<td>8 months ago</td>
</tr>
<tr>
<td>notebook</td>
<td></td>
<td>Fix #190</td>
<td>15 months ago</td>
</tr>
<tr>
<td>src/somemodule</td>
<td></td>
<td>update version</td>
<td>13 days ago</td>
</tr>
<tr>
<td>.gitignore</td>
<td></td>
<td>Fix test and added env to .gitignore</td>
<td>29 days ago</td>
</tr>
<tr>
<td>.readthedocs.yml</td>
<td></td>
<td>documentation</td>
<td>2 years ago</td>
</tr>
<tr>
<td>CITATION.cff</td>
<td></td>
<td>Add citation file</td>
<td>4 months ago</td>
</tr>
<tr>
<td>Dockerfile</td>
<td></td>
<td>updating Docker image</td>
<td>4 months ago</td>
</tr>
<tr>
<td>LICENSE</td>
<td></td>
<td>initial cleanup</td>
<td>2 years ago</td>
</tr>
<tr>
<td>README.md</td>
<td></td>
<td>update doc</td>
<td>13 days ago</td>
</tr>
<tr>
<td>config.json</td>
<td></td>
<td>Created script to generate models and updated python version to 3.9</td>
<td>8 months ago</td>
</tr>
<tr>
<td>mkdocs.yml</td>
<td></td>
<td>Fix #178</td>
<td>15 months ago</td>
</tr>
<tr>
<td>pyproject.toml</td>
<td></td>
<td>minor package changes</td>
<td>4 months ago</td>
</tr>
<tr>
<td>setup.py</td>
<td></td>
<td>Fix #437</td>
<td>28 days ago</td>
</tr>
</tbody>
</table>
Readme Analysis
- Supervised classification
- Regular expressions
- Header analysis

File exploration
- Notebooks
- Dockerfiles
- Documentation

GitHub API

SOMEF: Recognizing Metadata Categories

- Name (GA)
- Full title (RE)
- Description (SC, HA)
- Citation (SC, RE, HA)
- Installation instructions (SC, HA)
- Invocation (SC)
- Usage examples (HA)
- Documentation (HA, FE)
- Requirements (HA)
- Contributors (HA)
- FAQ (HA)
- Support (HA)
- License (GA, HA, FE)
- Stars (GA)

Method used (provenance):
- Supervised Classification (SC)
- Header Analysis and Synset comparison (HA)
- File Exploration (FE)
- Regular Expressions (RE)
- GitHub API (GA)
- Contact (HA)
- Download URL (HA, GA)
- DOI (RE)
- DockerFile (FE)
- Notebooks (FE)
- Executable notebooks (Binder, Collab) (RE)
- Owner: (GA)
- Keywords (GA)
- Source code (GA)
- Releases (GA)
- Changelog (GA)
- Acknowledgements (HA)
- Logos (RE)
- Images (RE)
- Shell scripts (FE)
- Code of conduct (FE)
- Repository status (RE)
- Arxiv links (RE)
- Support channels (RE)
- Software category (SC) (Work in progress)
- …
Converting containers and code into KGs

https://osoc-es.github.io/c2t/website/

https://github.com/SoftwareUnderstanding/inspect4py
Creating KGs of Research Software metadata: Benefits
Early result: Automated software catalogs

Useful for:
- Comparison
- Exploring
- Reuse

A software repository at a glance

{Morph-KGC: Scalable Knowledge Graph Materialization with Mapping Partitions}

Usage

Learn quickly with the tutorial in Google Colabary!
PyPI is the fastest way to install Morph-KGC:

```
pip install morph-kgc
```

We recommend to use virtual environments to install Morph-KGC.
To run the engine via command line you just need to execute the following:

```
python3 -m morph_kgc config.ini
```

Check the documentation to see how to generate the configuration INI file.

Here you can also see an example INI file.

It is also possible to run Morph-KGC as a library with RDFLib and Oxigraph:

```
import morph_kgc

# generate the triples and load them to an RDFLib graph
g_rdflib = morph_kgc.materialize("/path/to/config.ini")

# work with the RDFLib graph
q_res = g_rdflib.query("SELECT DISTINCT ?classes WHERE { ?s a ?classes }")

# generate the triples and load them to Oxigraph
q_oxigraph = morph_kgc.materialize("/path/to/config.ini")

# work with Oxigraph
q_res = graph.query("SELECT DISTINCT ?classes WHERE { ?s a ?classes }")

# the methods above also accept the config as a string
config = "...

[DataSource]

mappings: /path/to/mapping/mapping_file.rml

db_url: mysql+pymysql://user:password@localhost:...

```

```
g_rdflib = morph_kgc.materialize(config)
```

Citation

@article{arenas2022morph,
  title = {Morph-KGC: Scalable Knowledge Graph Materialization with Mapping Partitions},
  author = {Arenas-Guerrero, Julián and Chaves-Fraga, David and Toledo, David and Chaves-Fraga, Julián and Chaves-Fraga, David and Toledo, David and Chaves-Fraga, Julián and Chaves-Fraga, David and Toledo, David},
  journal = {Semantic Web},
  year = {2022},
  url = {http://www.semantic-web-journal.net/system/files/swj3135.pdf}
}
Analyzing community best practices

Extracting KGs from thousands of Open Source repositories
- Zenodo software (> 12000)
- Measuring best practices based on metadata
Summing up
Research software is a critical asset for **Open Science**
- Access information in structured, homogeneous manner
- Reusability
- Comparison
- Search

Pending **Challenges:**
- Automated metadata extraction from existing sources
- Curation
- Reconciliation of entities (through KGs like Wikidata)
- Representing all levels of granularity
Acknowledgements

Thanks to Yolanda Gil, Varun Ratnakar, Maximiliano Osorio, Hernán Vargas, Deborah Khider, Allen Mao, Aidan Kelley, Haripriya Dharmala, Jiajing Wang, Rosa Filgueira, Pablo Calleja, Oscar Corcho, Laura Camacho, Jhon Toledo, Miguel Angel García, Esteban Gonzalez, Elena Montiel, Elvira Amador & all the students at UPM and USC who participated in the initiatives mentioned in this presentation.

This work has been supported by the Madrid Government (Comunidad de Madrid-Spain) under the Multiannual Agreement with Universidad Politécnica de Madrid in the line Support for R&D projects for Beatriz Galindo researchers, in the context of the V PRICIT (Regional Programme of Research and Technological Innovation)
Let's create **machine-actionable** software metadata to promote Open Science!