Towards Assessing FAIR Research Software Best Practices in an Organization Using RDF-star

Ana Iglesias-Molina[1], Daniel Garijo[1]

[1]Ontology Engineering Group, Universidad Politécnica de Madrid, Spain

Abstract

An increasing number of scientists share the source code used or developed during their research (i.e., their research software) in open repositories, in order to support the results described in their publications. Recent best practices have been proposed by the community by aligning the Findable, Accessible, Interoperable and Reusable (FAIR) principles to Research Software. However, there are currently no means to assess the systematic adoption of these practices in a research organization. In this poster, we propose an automated pipeline to transform the software metadata of an organization as a Knowledge Graph in order to assess the current adoption of FAIR Research Software best practices. Our poster shows results from our own GitHub organization, the Ontology Engineering Group.

Keywords

Research Software, Metadata, FAIR software, FAIR, RDF-star

1. Introduction

Research Software (RS) [1] plays a crucial role in reproducing computational experiments, where it can range from simple visualization scripts or data cleaning libraries to complex computational pipelines that deliver the main findings described in a publication. RS has become key in many application domains, ranging from Astronomy[1] to Computational Biology [2]. Following the Findable, Accessible, Interoperable and Reusable (FAIR) principles for data [3] the scientific community has discussed and adapted FAIR to RS [1], making available guidelines and best practices for researchers and RS engineers [4, 5]. Unfortunately, while there are tools for helping developers adopt some of these practices [6], there is little work on assessing their overall adoption within a given organization.

In this poster we propose an automated pipeline that creates a Knowledge Graph (KG) of RS metadata to quantify the adoption of FAIR best RS practices within an organization. Our pipeline assesses online code repositories using existing software metadata extraction tools [7, 8] and combines them with RML-star mappings [9] to quantify the adoption of an illustrative set of best practices while tracking the provenance of each assertion.

The reminder of the paper first describes the best practices we focus on in Section 2, followed by the data model and mappings used to create our KG in Section 3. We show how we quantify best practices in Section 4, concluding the paper in Section 5.

SEMANTICS 2023 EU: 19th International Conference on Semantic Systems, September 20-22, 2023, Leipzig, Germany

ana.iglesiasm@upm.es (A. Iglesias-Molina); daniel.garijo@upm.es (D. Garijo)
0000-0001-5375-8024 (A. Iglesias-Molina); 0000-0003-0454-7145 (D. Garijo)

[1] https://www.ligo.caltech.edu/news/ligo20160211

Table 1
Ten key requirements to assess FAIR research software best practices of a code repository.

ID	Best practice	FAIR Principle	Source
BP1	A description (long or short) is available	F	[5, 4]
BP2	A persistent identifier (e.g., DOI) is available	F	[5, 4]
BP3	A download URL is available	A	[5, 4]
BP4	A semantic versioning scheme is followed	A	[4]
BP5	Usage documentation (including I/O) is available	I,R	[5, 4]
BP6	A license is declared	R	[5, 4]
BP7	An explicit citation is provided	R	[5, 4]
BP8	Software metadata (programming language, keywords, etc.) is available	F,R	[4]
BP9	Installation instructions are available	R	[4]
BP10	Software requirements are available	R	[4]

2. Metadata requirements for FAIR RS Best Practices

Table 1 shows a summary of the ten best practices we address in our work, based on existing guidelines for publishing FAIR RS [5, 4]. The list includes requirements for all principles, emphasizing those that rely on the metadata available in a code repository (e.g., availability of a description, license, keywords, citation, usage documentation, installation instructions). The list is a subset of [4], as i) our work is scoped towards code repositories and not external metadata registries, and ii) some of the guidelines are still under discussion within the community [4]. However, we consider this list as a first step towards assessing the adoption of FAIR best practices within a research organization.

While software metadata interchange vocabularies have been proposed by the scientific community (e.g., Codemeta[2]), there is no well adopted standard for describing software metadata within code repositories. Instead, different practices are followed for specific metadata or programming languages. For example, a common practice is to describe LICENSES is to generate a LICENSE file, although in some cases this information is found in a section of the README file, in a configuration file (e.g., pom.xml, setup.py) or in source code files. Similarly, the Citation File Format [10] has been increasingly adopted in thousands of research projects. However, it is still common to find references as Bibtex text within README files to indicate the publication associated with a software project. For these reasons, we consider key including the source of each assertion as part of our KG's metadata.

3. Creating a Research Software Metadata Knowledge Graph

Figure 1 shows an overview of the process we follow in order to extract and convert metadata from a set of repositories into RDF. In this case, we consider a GitHub organisation (oeg-upm) as input in order to access a set of repositories belonging to a research group or institution. We then filter those repositories of interest, e.g., by removing those that are work in progress, or adding external repositories of interest outside the target organisation. Next, we use the SOftware Metadata Extraction Framework (SOMEF) [7, 8] on each code repository, in order to

[2] https://codemeta.github.io/terms/

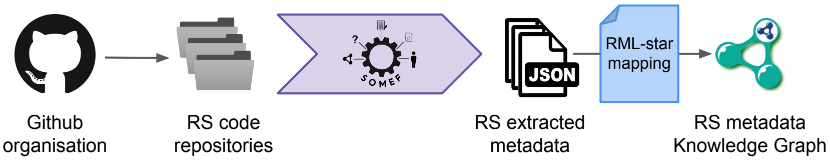


Figure 1: Overview of the worfklow to transform a set of repositories into a Knowledge Graph.

extract a structured and integrated metadata record. SOMEF extracts information from more than 40 metadata categories, keeping a clear provenance record of the source file where each metadata property was found (e.g., README, LICENSE.md, setup.py) the technique used in their extraction (e.g., supervised classification, regular expressions, analysis of the documentation headers) and the confidence (numeric value between 0 and 1) associated with the extraction. All metadata categories for a given software repository are included in a JSON file.[3]

Finally, we apply an RML-star mapping [9] with the Morph-KGC engine [11], in order to capture the confidence, techniques and source for each metadata property when building the Knowledge Graph. We rely on the Software Description Ontology (SDO) [12] to represent RS projects (**sd:Software)** their different releases (**sd:SoftwareVesion**s) and respective metadata. SDO extends Codemeta and Schema.org [13] for the Research Software domain, where both are gaining traction as a lightweight metadata interchange schema.

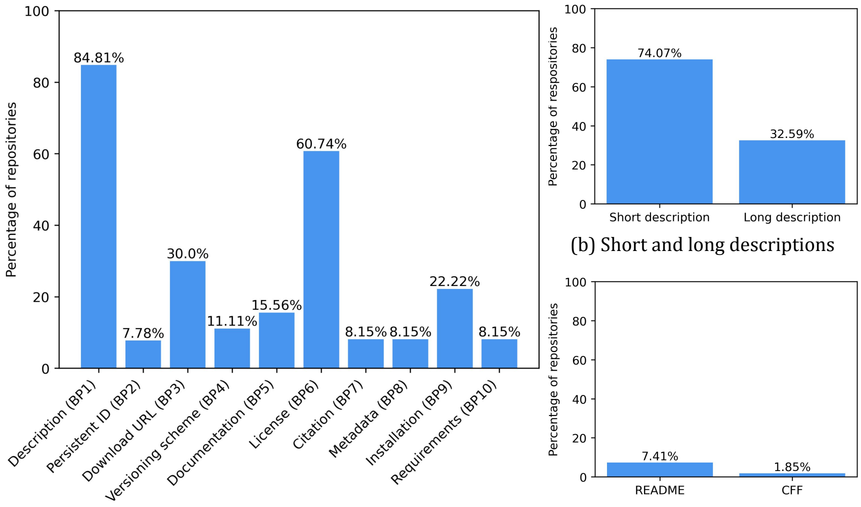
4. FAIR Research Software Best Practice Assessment

Figure 2 (a) shows the adoption of the best practices in Table 1 by 270 oeg-upm code repositories in GitHub. Figure 2 (b) shows additional insight in the type of descriptions provided on each repository. Long descriptions are paragraphs obtained from README files, and short descriptions are extracted from the *about* section in a GitHub repository page. Figure 2 (c) digs into the preferred methods for indicating a citation: within README files, in Citation File Format (CFF) files, or in both.

Overall, the results show a widespread adoption of short descriptions and licenses, with more than 60% of the repositories including them. The rest of the best practices are not as widely adopted yet. Downloading and installation guidelines are supported by up to 30% of the repositories, while independent documentation, metadata and citation are barely followed (less than 15% support). By looking at some of the repositories in detail, we notice that, in some cases, not all the best practices may apply. For instance, some repositories describe ontologies, and thus, have no need for installation instructions and requirements. The low number for other practices may be due to the additional effort required by researchers to accomplish them (e.g., providing independent documentation pages), or because the lack of awareness (e.g., CFF is followed by less than 2% of the repositories, showing a preference for inline README citations).

All figures have been generated by issuing SPARQL queries against a local Oxigraph instance [14] with support for RDF-star. Since metadata may vary depending on the extraction

[3] https://somef.readthedocs.io/en/latest/output/



(a) Compliance with Best Practices

(b) Short and long descriptions

(c) Citations per source

Figure 2: Overview with (a) the percentage of the 270 repositories compliant with each best practice, (b) zoom in repositories with short and long descriptions, and (c) main source of citations within repositories.

date, we use a named graph to store all triples in a given time stamp. For example, the following query assesses the number of citations in repositories which come from CFF files:

```
PREFIX sd: <https://w3id.org/okn/o/sd#>
PREFIX prov: <http://www.w3.org/ns/prov#>
SELECT (COUNT (DISTINCT ?software) AS ?count_software)
FROM <https://w3id.org/okn/i/graph/20230628>
WHERE {
    << ?software sd:citation ?cite >> prov:hadPrimarySource ?source
    FILTER(CONTAINS(str(?source),’.cff’))
}
```

Small edits to the query may be used to assess other metadata properties. All the data, mappings, KG, SPARQL queries and notebooks to run the pipeline are available online [15].[4]

5. Conclusions

In this poster we explore the FAIR RS best practices adoption within an organisation using an automated workflow. Our approach tracks the confidence, technique and provenance used in the extraction, enabling detailed queries and opening up the way towards designing validation mechanisms (e.g., SHACL shapes) and assistants to help researchers follow the FAIR principles.

Our future work aims at simplifying the extraction pipeline by integrating the RML-star mappings in SOMEF, thus expanding the best practice list with additional metadata, and addressing incomplete metadata in the extraction process.

[4] https://github.com/oeg-upm/oeg-software-graph

Acknowledgments

Ana Iglesias-Molina is supported by the project *Knowledge Spaces* (Grant PID2020-118274RB-I00 funded by MCIN/AEI/10.13039/501100011033). Daniel Garijo is 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 VPRICIT, and through the call Research Grants for Young Investigators from Universidad Politécnica de Madrid.

References

[1] N. P. Chue Hong, D. S. Katz, M. Barker, A.-L. Lamprecht, C. Martinez, F. E. Psomopoulos, J. Harrow, L. J. Castro, M. Gruenpeter, P. A. Martinez, et al., FAIR Principles for Research Software (FAIR4RS Principles), 2022. doi:10.15497/RDA00068.

[2] A. Prlić, H. Lapp, The plos computational biology software section, PLOS Computational Biology 8 (2012) 1–2. doi:10.1371/journal.pcbi.1002799.

[3] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L. B. da Silva Santos, P. E. Bourne, et al., The fair guiding principles for scientific data management and stewardship, Scientific data 3 (2016) 1–9. doi:10.1038/sdata.2016.18.

[4] M. Gruenpeter, S. Granger, A. Monteil, N. Chue Hong, E. Breitmoser, M. Antonioletti, D. Garijo, E. González Guardia, A. Gonzalez Beltran, et al., D4.4 - Guidelines for recommended metadata standard for research software within EOSC, 2023. doi:10.5281/zenodo.8097537.

[5] P. A. Martinez, C. Erdmann, N. Simons, R. Otsuji, S. Labou, R. Johnson, G. Castelao, B. V. Boas, A.-L. Lamprecht, C. M. Ortiz, et al., Top 10 fair data & software things, 2019. doi:10.5281/zenodo.3409968.

[6] J. H. Spaaks, S. Verhoeven, E. Tjong Kim Sang, F. Diblen, C. Martinez-Ortiz, E. Etuk, M. Kuzak, B. van Werkhoven, A. Soares Siqueira, S. Saladi, A. Holding, howfairis, 2022. URL: https://github.com/fair-software/howfairis.

[7] A. Kelley, D. Garijo, A Framework for Creating Knowledge Graphs of Scientific Software Metadata, Quantitative Science Studies (2021). doi:10.1162/qss_a_00167.

[8] A. Mao, D. Garijo, S. Fakhraei, Somef: A framework for capturing scientific software metadata from its documentation, in: 2019 IEEE International Conference on Big Data (Big Data), 2019, pp. 3032–3037. doi:10.1109/BigData47090.2019.9006447.

[9] T. Delva, J. Arenas-Guerrero, A. Iglesias-Molina, O. Corcho, D. Chaves-Fraga, A. Dimou, RML-star: A Declarative Mapping Language for RDF-star Generation, in: International Semantic Web Conference, P&D, volume 2980, CEUR Workshop Proceedings, 2021. URL: http://ceur-ws.org/Vol-2980/paper374.pdf.

[10] S. Druskat, J. H. Spaaks, N. Chue Hong, R. Haines, J. Baker, S. Bliven, E. Willighagen, D. Pérez-Suárez, O. Konovalov, Citation File Format, 2021. doi:10.5281/zenodo.5171937.

[11] J. Arenas-Guerrero, D. Chaves-Fraga, J. Toledo, M. S. Pérez, O. Corcho, Morph-KGC: Scalable knowledge graph materialization with mapping partitions, Semantic Web (2022). doi:10.3233/SW-223135.

[12] D. Garijo, M. Osorio, D. Khider, V. Ratnakar, Y. Gil, Okg-soft: An open knowledge graph with machine readable scientific software metadata, in: 2019 15th International Conference on eScience (eScience), IEEE, 2019, pp. 349–358.

[13] R. V. Guha, D. Brickley, S. Macbeth, Schema. org: evolution of structured data on the web, Communications of the ACM 59 (2016) 44–51.

[14] T. Pellissier Tanon, Oxigraph, 2023. URL: https://doi.org/10.5281/zenodo.8034456. doi:10.5281/zenodo.8034456, if you use this software, please cite it as below.

[15] A. Iglesias-Molina, D. Garijo, oeg-upm/oeg-software-graph: v1.0.0, 2023. doi:10.5281/zenodo.8114677.