



A Description Framework for Research Software and Metadata Publication Policies

David Pape ¹ // **Oliver Bertuch** ² // Tobias Huste ¹ // Oliver Knodel ¹ // Michael Meinel ³ // Sophie Kernchen ³ // Nitai Heeb ² // Christian Meeßen ⁴

¹ Helmholtz-Zentrum Dresden – Rossendorf (HZDR) // ² Forschungszentrum Jülich (FZJ) //
 ³ Deutsches Zentrum für Luft- und Raumfahrt (DLR) // ⁴ Helmholtz-Zentrum für Geoforschung (GFZ)

Software CaRD: Curation of Research Software Publications

github.com/softwarepub/ software-card-policies

team@software-metadata.pub



The curation of software metadata safeguards their quality and compliance with institutional software policies. Moreover, metadata that was development and enriched with usage information can be used for evaluation and reporting of academic KPIs. Software CaRD Reporting ("Software Curation and Dashboard"), a project funded by the Helmholtz Metadata Collaboration (HMC), develops tools to support the curation and reporting steps of the research software publication process. The

dashboard will present metadata collected by the HERMES workflow in a graphical user interface, assess compliance with a configurable set of policies, and highlight issues and breaches. It will be usable both standalone and in a CI/CD context.

As a first step in the project, and as a foundation for the curation dashboard, a description format for software publication policies had to be developed. Our solution

takes an approach that allows for configuration at different levels of abstraction: Low level building blocks describe metadata (e.g., CodeMeta) validation in terms of the Shapes Constraint Language (SHACL). A higher-level configuration language allows users to reuse and parameterize these components. This makes Software CaRD usable for RSEs, management, and policy makers, and it allows for customization that facilitates usage in different research institutions.

Policy Description Framework

Goal: Validation of software publication metadata (RDF, mainly using CodeMeta terms) produced by, e.g., HERMES.

<<u>https://orcid.org/0000-0002-1825-0097</u>> a <u>schema:</u>Person ;

- schema:name "Josiah Carberry";
- schema:email "don't have one" ;
- schema:affiliation [
- a schema:Organization ;
- schema:legalName "Brown University"



····]·.

ex:simulationSoftware a schema:SoftwareSourceCode ;
 schema:codeRepository "<u>https://example.com/code/simulation.git</u>"^^schema:URL ;
 schema:description "My code. Is this enough of a description??" ;
 schema:license "UNLICENSE" ;
 schema:version "proof-of-concept" ;

schema:author <https://orcid.org/0000-0002-1825-0097> .

Does this software publication adhere to the publication policy of Josiah's university?

config.toml

[policies.licenses]

suggested_licenses = [

[policies.licenses.parameters]

source = "http://127.0.0.1:8000/policies/licenses.ttl'

"https://spdx.org/licenses/GPL-3.0-or-later",

"https://spdx.org/licenses/CC-BY-SA-4.0",

Parameterization Approach:

policies/licenses.ttl

- ex:suggestedLicenses a sc:Parameter ;
 sc:parameterType rdf:List ;
 sc:parameterConfigPath "suggested_licenses" ;
- sc:parameterDefaultValue (
- "https://spdx.org/licenses/Apache-2.0"
- "https://spdx.org/licenses/MIT"
- ••••)••
- ex:licenseRequirements a sh:NodeShape ;
- sh:targetClass schema:SoftwareSourceCode ;
- •••<mark>sh:</mark>property [
- sh:name "Suggested license";
- sh:description "A license from this list must be chosen.";
- sh:path schema:license;
- sh:datatype xsd:string;
- sh:in ex:suggestedLicenses;

····]·.

Policy Creation Workflow:







Software CaRD is embedded into the automated "HERMES" publication workflow* "curation" step. Using the same Continuous Integration runner provides the same access and interaction options. *) https://hermes.software-metadata.pub

Policy Examples Authors' affiliations must include a ROR Authors must use their institutional email address Description texts must be at least 300 characters long The software version should adhere to semantic versioning The source code repository on the institutional forge must be linked

Community Involvement

A work-in-progress Python implementation of the framework, using RDFLib and pySHACL, is available on GitHub. Your feedback on

Reusable, configurable policies (SHACL shapes and constraint components) Policy packages specific to institutional requirements (as e.g. SHACL, Python packages)

Reuse packages, override configuration

specification, use cases, and collaboration are welcome!





softwarepub/**software**card-policies



Collection of example policies and tools developed in the Software CaRD project.

용 1 ⓒ 21 ☆ 0 ♀ 0 Contributor Issues Stars Forks

 \bigcirc

Validation Report: Based on SHACL validation report (sh:ValidationReport)

Curator Report

Which policies were breached?

Did the user override any parameters?

<u>User Report</u>

Which policies were breached?

Who wrote these policies?

What are the policies for?

Acknowledgments

Software CaRD (ZT-I-PF-3-080) is funded by the Initiative and Networking Fund of the Helmholtz Association in the framework of the Helmholtz Metadata Collaboration.



