A metadata schema for machine-actionable Software Management Plans

Olga Giraldo[1] [0000-0003-2978-8922], Lukas Geist[1] [0000-0002-2910-7982], Nelson Quiñones[1], Dhwani Solanki[1], Renato Alves[2], Dimitrios Bampalikis[3], José M. Fernández[4] [0000-0002-4806-5140], Eva Martin del Pico[4], Fotis Psomopoulos[5] [0000-0002-0222-4273], Allegra Via[6] [0000-0002-3398-5462], Dietrich Rebholz-Schuhmann[1,7] [0000-0002-1018-0370], Leyla Jael Castro[1 *] [0000-0003-3986-0510]

[1] ZB MED Information Centre for Life Sciences, Cologne, Germany
[2] European Molecular Biology Laboratory, Heidelberg, Germany
3 National Bioinformatics Infrastructure Sweden, Uppsala, Sweden
[4] Barcelona Supercomputing Center, Barcelona, Spain
[5] Centre for Research and Technology Hellas, Thessaloniki, Greece
[6] Institute of Molecular Biology and Pathology - National Research Council, Rome, Italy
[7] University of Cologne, Cologne, Germany
* ljgarcia@zbmed.de

Abstract. Data-driven research requires handling data together with the software that is used to collect, transform, and create such data. Data Management Plans have emerged as a systematic way to record the data management lifecycle for data corresponding to a research project. Similar to DMPs, Software Management Plans (SMPs) follow the research software management lifecycle, becoming a complement of DMPs. Initially, both DMPs and SMPs were conceived as text-based documents, sometimes guided by a set of questions targeting key points related to the corresponding lifecycle. Machine-actionable DMPs improve text-based DMPs by adding a semantic layer representing the most common elements relevant to DMPs, from datasets to funders. Here, we use the ELIXIR SMP as a use-case and present a preliminary metadata schema including possible types and properties useful to represent machine-actionable SMPs.

Keywords: Research software, Management Plan, Metadata, machine-actionable

1 Background

Recognition of the role that software plays on data-driven research projects has gained importance in recent years as software is fundamental for collecting, transforming and combining data. This has led to an increasing number of software-based solutions used or developed in research laboratories and institutions [1]. Similar to other research methods, research software must be documented, published, and acknowledged. Information about the processes and tools used to manage version control, and the licenses used when publishing software are key in the ***access*** and ***reuse*** of research software, essential elements of open science.

Data Management Plans (DMPs) are an important element that follows open science practices. A DMP describes the data management lifecycle for the data to be

collected, processed and/or generated within the lifetime of a particular project or activity [2]. DMPs are commonly text-based documents available in multiple formats not necessarily machine-readable. A new generation of machine-actionable DMPs (maDMPs) was therefore proposed by the Research Data Alliance DMP Common Standards Working Group to enable automated integration of information and updates [3]. maDMPs make use of persistent identifiers and a semantic layer [4] to connect all resources associated with a DMP.

People working with research software found that some information related to the management of the software could not be properly documented in a DMP. To cover that limitation, different groups have come up with proposals for (research) Software Management Plan (SMP). For instance, The Software Best Practices Focus Group, part of the ELIXIR Tools Platform, proposed an SMP for Life Sciences to provide a clear management context for research software [2]. Similarly, the Netherlands eScience Center and the Dutch Research Council (NWO) took the initiative to form a working group and develop (national) guidelines for domain-agnostic SMPs [5]. SMP templates commonly include a set of requirements and questions to ensure that researchers consistently adhere to certain software management standards and policies when developing research software.

As text-based documents, SMPs are not directly machine-readable nor do they support structured metadata. To overcome this issue, we propose to create a metadata schema extending the application profile defined for maDMPs together with the DMP Common Standard ontology (DCSO), with terminology related to research software. An application profile corresponds to a set of metadata elements together with some guidelines on how to use them wrt to a particular application. Our first version or a metadata schema for machine-actionable SMP (maSMP) contains metadata elements but does not yet offer guidelines or recommendations on how to use it for a particular use case. However, it is flexible enough, same as the maDMP application profile, to facilitate its extension and customization. Our maSMP metadata schema will allow us to i) obtain FAIR machine-actionable metadata tailored to software, and ii) facilitate metadata exchange across platforms and funders. Here we present our preliminary results.

2 Machine-actionable Software Management Plans

We propose to extend DCSO, with terminology related to SMPs, taking the one proposed by ELIXIR as the baseline. To achieve this goal, we defined a roadmap, which is presented below. Later, we present our preliminary results, mainly corresponding to the first draft of the maSMP metadata schema.

maSMP roadmap

Our roadmap includes the following stages: (i) conceptualization of SMP elements, (ii) semantic analysis of existing vocabularies, (iii) ontology building and validation, (iv) mapping to schema.org and Bioschemas, and (v) evaluation. These stages are illustrated in Fig. 1 and explained below.

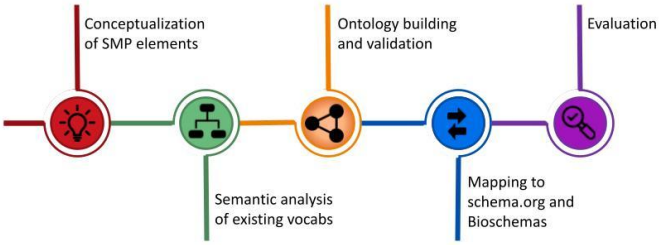


Fig. 1. maSMP roadmap.

I. **Conceptualization of SMP elements:** Gather a list of data elements related to documentation, testing, interoperability, versioning, reproducibility and recognition of the software, based on the ELIXIR SMP questionnaire. The list of data elements is shared with domain experts (computer scientists and bioinformaticians), they review the drafts, give feedback and the list is updated.

II. **Semantic analysis:** Identify and select ontologies that could be reused in order to add semantics to the data elements proposed in the first stage. The idea of having a metadata schema in the form of an ontology represented in OWL is to explore the potential of a semantic layer behind a machine-actionable version of the Software Management Plan template available in the SM Wizard [6].

III. **Ontology building and validation:** Take into account metadata elements from the maDMP application profile and corresponding ontology, DCSO, in order to facilitate integration (interconnection or extension) of our ontology module. The ontology drafts are shared with domain experts, they review the drafts, give feedback and the ontology is updated.

IV. **Mapping to schema.org and Bioschemas:** Facilitate linked data to external resources like schema.org [7, 8]. The mapping to schema.org will follow the approach proposed by Bioschemas [9, 10].

V. **Evaluation:** Address issues related to syntax (i.e., identification of incomplete inverse object properties, lack of domain and range, missing annotations), conceptualization and formalization (i.e., determine if the proposed classes in the ontology represent the information recommended by the ELIXIR SMP).

Preliminary results

I. **Conceptualization of SMP elements:** We have created a first draft of a data elements list related to documentation, testing, interoperability, versioning, reproducibility and recognition of the software. The output of this activity was the first version of a checklist available at [11]. Valuable feedback was received from domain experts and the list is being updated.

II. **Semantic analysis and ontology building:** We have defined a metadata schema in the form of an ontology representing the necessary metadata elements for a maSMP. The metadata schema includes entities involved in software management planning; such as an SMP itself, software source code, software release, documentation, authors and their relations. We are reusing terms mainly from schema.org, Bioschemas and from DCSO, with some few additions of our

own. An overview of concepts used in the metadata model for maSMPs is available at [12] and depicted in Fig. 2.

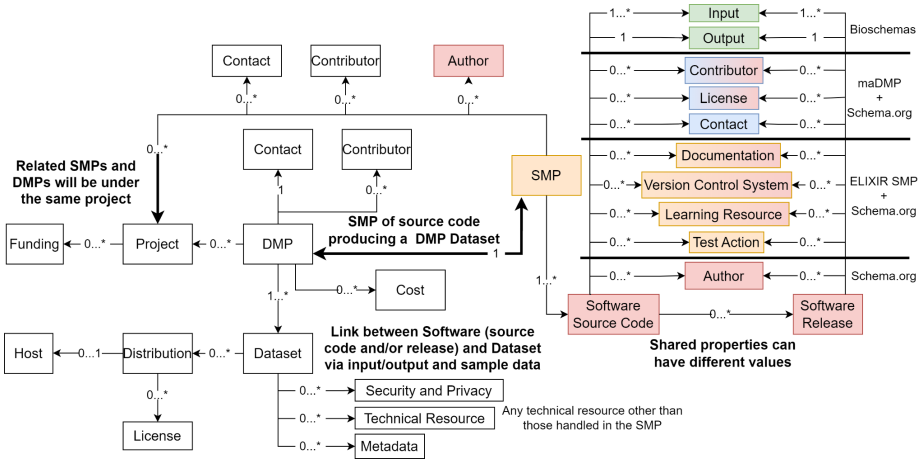
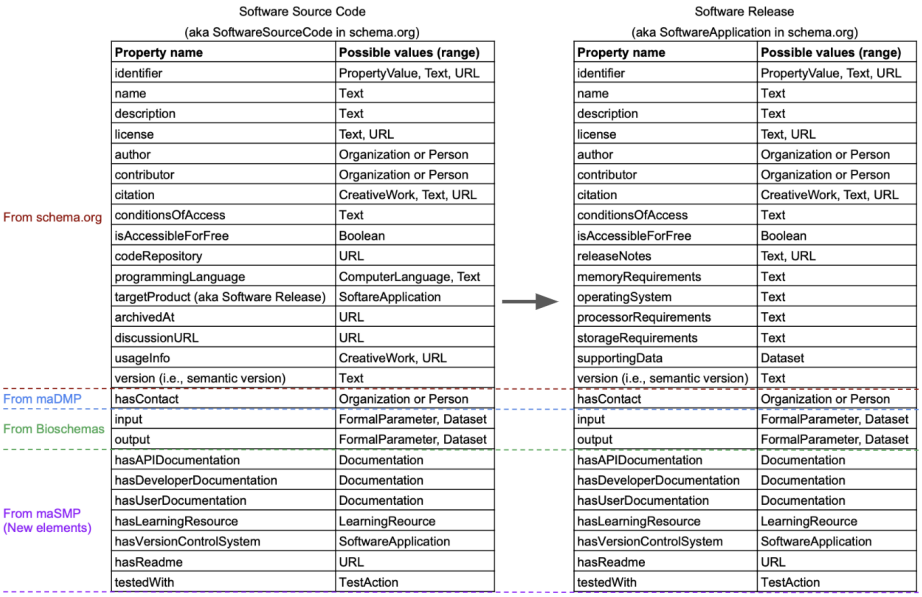


Fig. 2. SMP metadata model, and its connections to the maDMP application profile. Boxes with colored background correspond to the elements added for the maSMP case.

Software Source Code and Software Release share most of the object properties (i.e., those that point to another object rather than to a simple type such as a number) but they correspond to different software concepts. While the source code reflects the current status of a software and can be continuously changing, a software release corresponds to a frozen copy of a particular version. As the source code evolves, shared elements can differ, e.g., new authors can get involved. This changing nature of source code together with the release cycle are important aspects captured by the maSMPs that otherwise might not be evident in text-based questionnaires. In Fig. 3 we show a closeup corresponding to these two elements as they are the backbone of maSMPs. In the figure, we include the internal elements which correspond to a metadata representation of the questions asked by the ELIXIR SMP. Some elements present in the source code but not in the release are, for instance, a discussion URL to collect hugs and bugs, as well as a pointer to a live registry or archival, e.g., the Software Heritage Archive [13, 14]. Releases are also expected to be deposited in a software repository, where they should get a persistent identifier that can be used to reference and cite them. The first released version of the ontology is available at [15]; however, as it is still work-in-progress at preliminary stages, it does not correspond yet to a stable version.



Software Source Code
(aka SoftwareSourceCode in schema.org)

	Property name	Possible values (range)
From schema.org	identifier	PropertyValue, Text, URL
	name	Text
	description	Text
	license	Text, URL
	author	Organization or Person
	contributor	Organization or Person
	citation	CreativeWork, Text, URL
	conditionsOfAccess	Text
	isAccessibleForFree	Boolean
	codeRepository	URL
	programmingLanguage	ComputerLanguage, Text
	targetProduct (aka Software Release)	SoftareApplication
	archivedAt	URL
	discussionURL	URL
	usageInfo	CreativeWork, URL
	version (i.e., semantic version)	Text
From maDMP	hasContact	Organization or Person
From Bioschemas	input	FormalParameter, Dataset
	output	FormalParameter, Dataset
From maSMP (New elements)	hasAPIDocumentation	Documentation
	hasDeveloperDocumentation	Documentation
	hasUserDocumentation	Documentation
	hasLearningResource	LearningReource
	hasVersionControlSystem	SoftwareApplication
	hasReadme	URL
	testedWith	TestAction

Software Release
(aka SoftwareApplication in schema.org)

Property name	Possible values (range)
identifier	PropertyValue, Text, URL
name	Text
description	Text
license	Text, URL
author	Organization or Person
contributor	Organization or Person
citation	CreativeWork, Text, URL
conditionsOfAccess	Text
isAccessibleForFree	Boolean
releaseNotes	Text, URL
memoryRequirements	Text
operatingSystem	Text
processorRequirements	Text
storageRequirements	Text
supportingData	Dataset
version (i.e., semantic version)	Text
hasContact	Organization or Person
input	FormalParameter, Dataset
output	FormalParameter, Dataset
hasAPIDocumentation	Documentation
hasDeveloperDocumentation	Documentation
hasUserDocumentation	Documentation
hasLearningResource	LearningReource
hasVersionControlSystem	SoftwareApplication
hasReadme	URL
testedWith	TestAction

Fig. 3. Detailed view of Software Source Code and Software Release as proposed in our metadata schema, including indication on the provenance of the properties

3 Future work

We will continue the evaluation and validation of our set of data elements. Workshops/meetings are already planned with domain experts in order to better align our list of data elements to current SMPs (e.g., [2] and [5]) and metadata schema for research software (e.g., Bioschemas and CodeMeta [16]). In parallel we will continue the development of the ontology/metadata schema together with examples of use. The ontology will represent the set of metadata evaluated and validated by domain experts. Then, we will start with the Bioschemas specification for maSMPs. The Bioschemas specification will not cover all the elements involved in the metadata schema for SMPs, but the main ones describing the source code and releases.

Our metadata schema represented as an ontology is still a work in progress. Machine-actionability for SMPs is not a single event, it is a team effort that brings together everyone related to the management of the research software, scholarly adoption and use. We plan to collaborate with other communities addressing the issues of research software management to align efforts to make broad adoption and interoperability across options easier.

Acknowledgments

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101017536 and is part of the Research Data Alliance and European Open Science Cloud Future call 2022.

References

1. Scheliga KS, Pampel H, Konrad U, Fritzsch B, Schlauch T, Nolden M, et al. Dealing with research software: Recommendations for best practices. 2019. https://doi.org/10.2312/os.helmholtz.003
2. Alves, R., Bampalikis, D., Castro, L., Fernández, J. M., Harrow, J., Kuzak, M., … Via, A. (2021, October 25). ELIXIR Software Management Plan for Life Sciences. https://doi.org/10.37044/osf.io/k8znb
3. Miksa T, Oblasser S, Rauber A. Automating Research Data Management Using Machine-Actionable Data Management Plans. ACM Trans Manage Inf Syst. 2021;13: 18:1-18:22. https://doi.org/10.1145/3490396
4. Miksa, T., Walk, P., & Neish, P. RDA DMP Common Standard for Machine-actionable Data Management Plans (Version 1.1) [Computer software]. https://doi.org/10.15497/rda00039
5. 1. Martinez-Ortiz C, Martinez Lavanchy P, Sesink L, Olivier BG, Meakin J, de Jong M, et al. Practical guide to Software Management Plans. Zenodo; 2022 Oct. https://doi.org/10.5281/zenodo.7248877
6. SM Wizard. Available at https://smw.ds-wizard.org/
7. Schema.org. Available at https://schema.org
8. Guha RV, Brickley D, Macbeth S (2016) Schema.org. Communications of the ACM 59 (2): 44-51. https://doi.org/10.1145/2844544
9. Gray AJG, Goble C, Jimenez RC (2017) From Potato Salad to Protein Annotation. ISWC Posters and Demo session. URL: http://ceur-ws.org/Vol-1963/paper579.pdf
10. Bioschemas Governance. Available at https://github.com/Bioschemas/governance/blob/master/governance.md
11. Castro LJ, Quiñones N, Bampalikis D, Giraldo O, Del Pico EM, Via A, et al. A metadata analysis for machine-actionable Software Mng Plans - Poster. ZB MED - Informationszentrum Lebenswissenschaften; 2023. Available: https://repository.publisso.de/resource/frl:6440396 https://doi.org/10.4126/FRL01-006440396
12. Giraldo O., Geist L., Quiñones N., Solanki D., Rebholz-Schuhmann D., and Castro LJ. Machine-actionable Software Management Plans (Version 0.0.1) [Dataset]. https://github.com/zbmed-semtec/maSPMs
13. Software Heritage Foundation. Software Heritage Archive. Available at https://archive.softwareheritage.org/
14. Abramatic J-F, Di Cosmo R, Zacchiroli S. Building the universal archive of source code. Commun ACM. 2018;61: 29–31. https://dl.acm.org/doi/10.1145/3183558
15. Giraldo O., Geist L., Quiñones N., Solanki D., Rebholz-Schuhmann D., and Castro LJ. Machine-actionable Software Management Plan Ontology (maSMP ontology) (Version 0.0.1) [Dataset]. https://doi.org/10.5281/zenodo.7806639
16. Boettiger C. et al. CodeMeta: Minimal metadata schemas for science software and code, in JSON-LD. (Version 2.0). Available at https://github.com/codemeta/codemeta