Electronic Communications of the EASST
Volume 83 Year 2025

deRSE24 - Selected Contributions of the 4th Conference for Research Software Engineering in Germany

Edited by: Jan Bernoth, Florian Goth, Anna-Lena Lamprecht and Jan Linxweiler

Requirements for a Metadata Scheme to Enable FAIR Energy Research Software

Stephan Ferenz, Oliver Werth, Astrid Nieße

DOI: 10.14279/eceasst.v83.2594

Electronic Communications of the EASST (https://eceasst.org).
Published by **Berlin Universities Publishing**
(https://www.berlin-universities-publishing.de/)

Requirements for a Metadata Scheme to Enable FAIR Energy Research Software

Stephan Ferenz[1 2], **Oliver Werth**[2], **and Astrid Nieße**[1 2]

[1] stephan.ferenz@uol.de, astrid.niesse@uol.de
Department for Computer Sicence
Carl von Ossietzky Universität Oldenburg, Germany

[2] oliver.werth@offis.de
Energy Division
OFFIS, Germany

Abstract: Energy Research Software (ERS) plays a significant role in energy research, by aiding with visualization, simulation, and analysis. To enhance its efficiency, metadata are crucial for better findability, accessibility, interoperability, and reusability (FAIR) of research software. The current approaches to metadata for ERS have multiple limitations, e.g., they are not based on a systematic approach to include the diverse requirements of energy researchers. To address this issue, a qualitative study was conducted to gather specific requirements for metadata for ERS. The findings show the need for additional metadata elements for research software, e.g., on the community or the support options. Also, domain-specific metadata are required, e.g., on time and geographical scope of an ERS. Subsequently, a domain model was developed based on the requirements that lays the foundation for creating a metadata scheme for ERS.

Keywords: Metadata Scheme, Metadata Schema, Energy Research Software, Information Requirements

1 Introduction

Energy Research Software (ERS) is software used in the scientific discovery process for understanding, analyzing, improving, and designing energy systems. ERS can be found at different levels of complexity, from basic scripts or libraries, e.g., in Python, to complete software solutions. In terms of content, ERS may involve visualization, analysis, and generation of (artificial) data associated with energy systems[1], components, or grids (from labs or the real world). Moreover, ERS can also represent specific energy components, systems, and their transition paths[2] in terms of energy usage, distribution, conversion, and generation for evaluations in simulations and

[1] An energy system consists of components for energy generation (e.g., power plants), conversion (e.g., transformers), distribution (e.g., network for electricity (grid), gas, heat), and usage (e.g., households, industry).

[2] Transition paths model how the energy system of a certain region (e.g., Germany, Europe) can transform over the next years. This is, for example, used to analyze which changes are required to achieve a climate neutral energy system by 2045 in Germany.

optimizations. [FN23] By providing theses various functionalities, ERS heavily supports research in the energy domain and presents the foundation for different research results in energy research.

ERS is facing multiple challenges, e.g., through the increasing complexity of energy systems[3] and of research itself [FN23]. Metadata have been shown to be one of the success factors for the so-called FAIRification of research software, especially to improve findability and reusability of research software [CKB+22, KGH21, BCK+22]. To reach a high level of findability, metadata should follow domain-relevant community standards [WDA+16]. Therefore, the development of such domain-relevant metadata standards presents a key prerequisite to enable FAIR ERS [FN23].

In [FN23] an overview of the few existing approaches to metadata for ERS was already given. None of these approaches use a formalized and interoperable metadata scheme to open the approach for further reuse for FAIR ERS. Also, none of the approaches are based on a requirement analysis, which is important to identify meaningful metadata elements [CB13]. A requirement analysis should be the first step in developing a metadata scheme to identify the information that should be included in the metadata [CB13][4]. Based on the requirements, it can be analyzed if an existing scheme can be extended or which elements from existing schemes can be reused. Therefore, the goal of our work is to gather these specific requirements for a metadata scheme for ERS to answer the following research question (RQ):

RQ: What are the requirements for metadata of ERS that support energy researchers in finding and/or selecting ERS for reuse?

With this paper, we contribute as follows:

- We summarize the state of the art in the field of metadata schemes for ERS and related approaches as well as their way to collect requirements in Section 2.
- Based on the method, data collection, and analysis, described in Section 3, we present the gathered requirements for metadata of ERS in Section 4.
- We discuss our analysis and give an outlook of the further required work in Section 5.

2 Related Work

Within this section, we give an overview of existing metadata schemes for research software (based on [FN23]) and the method they used to gather requirements for the schemes. First, we have a look at general approaches and approaches from other domains in Subsection 2.1. Afterwards, we give an overview of surveys which included questions on information requirements when searching for research software in Subsection 2.2. As the last part, we present approaches from the energy domain in Subsection 2.3.

[3] Energy systems are changing through digitalization and the energy transition at the same time.

[4] Instead of developing a completely new metadata scheme, another option is to develop an application profile. An application profile recombines and refines metadata elements from other existing metadata schemes [GMB+22] to address a specific application. Nevertheless, the requirements will be the same for both approaches. Therefore, we will use the term metadata scheme for the rest of this paper.

Table 1: Overview of metadata schemes for research software (based on [FN23])

	Metadata scheme (s) vs. ontology (o)	Domain	Methods for requirements
CodeMeta [JBM+17]	s	General	Based on existing approaches
OntoSoft [GRG15]	o	Geoscience	Informal surveys
Software Desciption Ontology [GOK+19]	o	Geoscience	Source of requirements unclear
Software Ontology [MBL+14]	o	Bioinformatics	Requirements workshop
biotoolsXSD [IIR+21]	s	Bioinformatics	Community workshops

2.1 General Approaches and Approaches from Other Domains

The general approaches and approaches from other domains are summarized in Table 1.

CodeMeta[5] [JBM+17] is a community-driven metadata standard for research software based on schema.org[6]. It was developed based on existing approaches from different repositories[7]. Further additions are discussed on GitHub.

For geosciences, Gil et al. introduced an ontology to describe research software, *OntoSoft*, with six categories: identify, understand, execute, do research, get support, and update [GRG15]. They reported that they performed informal surveys [GGMR16] without providing more details on their method.

Garijo et al. expanded this approach by developing the *Software Description Ontology*[8] [GOK+19] with additional descriptions for input and output data based on the *Scientific Variables Ontology*[9]. They used competency questions to develop the *Software Description Ontology* without providing a further source for these.

The *Software Ontology (SWO)* was developed by extending the bioinformatics *EDAM* ontology to describe software in this research field. The requirements were gathered from stakeholders from different domains (archiving organization, software sustainability, library services, astronomy, life science, and pharmaceutical research) based on user stories and competency questions in face-to-face workshops. In two additional workshops, the ontology was improved. [MBL+14]

Also for bioinformatics, Ison et al. developed the metadata scheme *biotoolsXSD* for the software registry bio.tools[10]. The requirements for the scheme were derived from multiple community workshops. [IIR+21]

[5] https://codemeta.github.io/, accessed 26.03.2024
[6] https://schema.org/, accessed 26.03.2024
[7] https://github.com/codemeta/codemeta, accessed 26.03.2024
[8] https://w3id.org/okn/o/sd, accessed 26.03.2024
[9] https://scientificvariablesontology.org/svo/, accessed 12.12.2022
[10] http://bio.tools, accessed 26.03.2024

Table 2: Overview of metadata schemes for ERS (based on [FN23])

	Formalized metadata scheme?	Methods for requirements
Catalog of energy co-simulation components [SL19]	partly fulfilled	No information provided
openmod[12]	not fulfilled	No information provided
Open Energy Platform factsheets on models[13]	not fulfilled	No information provided

2.2 Research on Requirements for Metadata for Research Software

Besides the research towards specific metadata schemes, there is also some more general research on information requirements for (research) software catalogs.

Hucka and Graham conducted an online survey with 69 participants from different scientific domains. They looked for information scientists like to find in a software catalog and into criteria for choosing software to reuse. For both aspects they let the participants choose from predefined answers. They identified special features as the most important selection criteria. [HG18]

Stevens conducted a similar survey as Hucky and Graham with an improved set of questions. He had 156 responses to his survey. In this survey, the availability of special features was also the most important characteristic when looking for research software. [Ste22]

Both surveys focused on researchers from multiple domains and are important for statistical analysis of important characteristics for research software when searching. Since they used predefined answers, they were not able to explore additional relevant characteristics of research software, which we would like to do in this study.

2.3 Energy-related Approaches

As the second step, we have a look at approaches from the energy domain which are summarized in Table 2.

Schwarz and Lehnhoff described a catalog of energy co-simulation components[11]. They did not provide further information on the development of this catalog. The elements of the catalog are usable as a metadata scheme but are neither formalized nor described in more detail. [SL19]

The Open Energy Modeling Initiative (openmod) includes a list of energy models in their wiki[12]. The metadata scheme is not formalized and controlled vocabularies are used neither for the elements nor the values. There is no information given regarding if and how requirements for the metadata were gathered.

Additionally, the Open Energy Platform[13] includes information on models and frameworks. The metadata elements are similar to the ones of the openmod wiki and also not formalized. There is no information provided about if and how requirements for the metadata were gathered.

[11] Co-simulation is a specific type of simulation where the optimization and models are distributed. This is especially interesting when analyzing the operation of energy systems.
[12] https://wiki.openmod-initiative.org/wiki/Open_Models, accessed 12.12.2022
[13] https://openenergy-platform.org/factsheets/models/, accessed 26.03.2024

Table 2 shows that, as far as it is documented, none of the existing approaches for metadata for ERS are based on a requirement analysis, which is a gap we would like to close with this publication. Different approaches for collecting requirements were chosen in the past for metadata schemes for general research software and research software in other domains as can be seen in Table 1. Most of the approaches give little or no detailed description of their method for their requirement analysis. Stevens and Hucka and Graham used online surveys for their analysis. In contrast to that, we would like to openly explore the requirements for a metadata scheme for ERS by conducting interviews. Additionally, we investigate domain-specific requirements for the energy domain.

3 Research Design and Method, Data Collection and Analysis

In this research, we aim to explore the requirements for metadata for ERS, as formulated in our RQ in Section 1. To identify the relevant requirements, we utilized a qualitative approach, engaging in expert interviews to gather opinions and thoughts from various stakeholders.

We designed a semi-structured interview guideline based on the initial research question and the FAIR principles for research software [BCK+22] to conduct the interviews with an appropriate level of reliability [Sil16]. The guideline ensured flexibility, allowing interviewees to express their information needs regarding ERS. Additionally, we promised confidentiality of the interview transcripts to avoid possible response biases [MN07]. The interview guideline[14] was structured into seven parts, beginning with starting questions about the research of the interviewees and their use of research software. Afterward, we oriented ourselves on five question blocks about different information usages of metadata for ERS based on the FAIR principles [BCK+22]: findability from a search perspective, findability from a selection perspective, accessibility, interoperability, and reusability. Finally, there were some concluding questions, e.g., on the general opinion on a research software registry in the energy domain. We performed a pretest interview to validate the guideline with a Ph.D. student. Next, we adapted the interview guidelines based on the experience from this first interview.

To find participants for the interviews, we considered experts from various stakeholder groups, each contributing to different research aspects in the energy domain, e.g., research on energy systems, and research on components for energy systems. We focused on experts with experience with research software. We aimed to include a diverse range of researchers, including those at different career levels, such as Ph.D. students, postdoctoral researchers, and professors, as well as male and female researchers. By selecting a diverse range of researchers, we reduced the effects of biases of single interviewees. In respect to career levels we aimed at an equal distribution to avoid response biases [NS15].

Also, we included industry experts to incorporate their perspectives on metadata of ERS. We identified prospective interviewees through the researchers' networks and invited them via email. In total, we conducted 32 expert interviews with four stakeholder groups between August 2022 and January 2023. Figure 1 and Table 3 give an overview of the distribution of the interviewees. The interviews were held in German[15] or English, in-person or online, and lasted between 20 and

[14] Available in English and German at https://doi.org/10.5281/zenodo.11189410
[15] When citing German interviews in Section 4, we only give the translation of the citation.

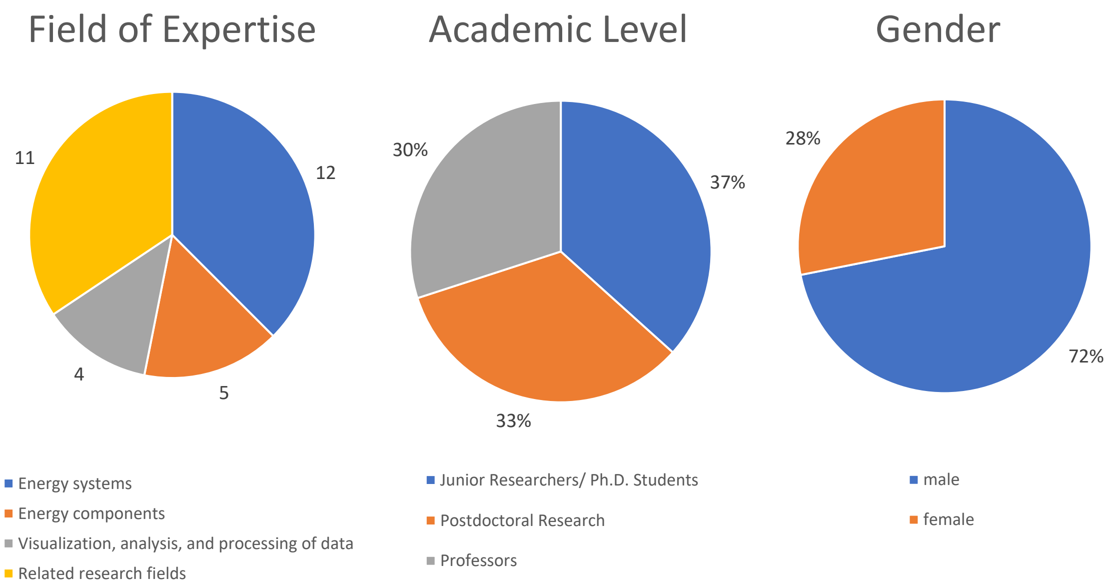

Figure 1: Profiles of Interviewees

Stakeholder Groups		Junior Researcher/ Ph.D. Students	Senior Researcher/ Group Leaders	Professors	Other
	Number of Interviews	11	10	9	2
Visualization, analysis, and processing of data	4	I19, I21, I30		I20	
Energy systems	12	I7, I11, I13, I18, I26	I10, I12, I29, I17, I24	I27, I31	
Energy components	5	I2, I5	I4, I8		I16
Related research fields (e.g., heat, energy economics, social sciences, ...)	11	I1	I3, I23, I32	I6, I9, I14, I15, I25, I28	I22

Table 3: Profiles of Interviewees

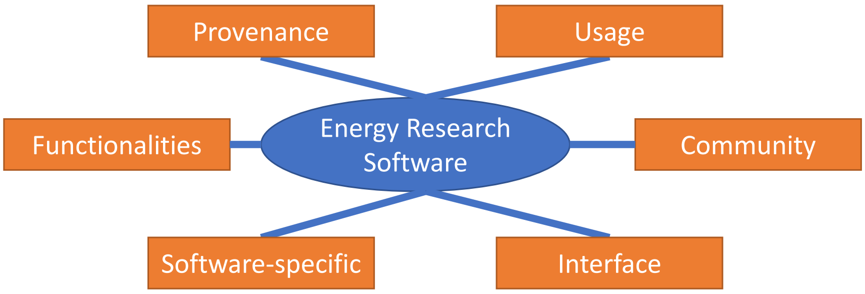

Figure 2: General Domain Model - First-Order Codes

80 minutes. We achieved theoretical saturation after 32 interviews as new content and concepts did not emerge [SC90]. All interviews were transcribed using *whisper*[16].

The primary data analysis involved qualitative content analysis [GS17] using *MAXQDA 2022 Analytics Pro software*[17].

Initially, we examined all available data, comparing it for common themes or similarities and assigning first-order codes based on information areas of ERS, such as ”community”. By examining these first-order codes, we identified particular information needs that closely resembled potential metadata elements, e.g., ”existing community interactions”. Then, we gathered and consolidated these findings into themes which we labeled with second-order codes, e.g., ”contributor”. These themes and potential metadata elements have been organized into a domain model, which is presented in Section 4.

This domain model represents the result of our qualitative content analysis, providing a structured framework for understanding the information requirements for ERS in the context of our study. For the definitions of the metadata elements, we reused *CodeMeta* as much as possible. To ensure consistency we added a few elements between the elements mentioned in the interviews.

4 Requirements for Metadata of Energy Research Software

Within this section, we give an overview of the requirements for metadata of ERS resulting from the expert interviews. As described in Section 3, the requirements were directly developed into a domain model. We identified six first-order codes which are the top categories of our domain model, as shown in Figure 2, and which will serve as subsections: provenance (4.1), usage (4.2), community (4.3), interface (4.4), software-specific (4.5), and functionalities (4.6). In the following, we give more details on each of them.

4.1 Provenance

The category ”provenance” focuses on information about the origin of the software. The different identified terms are summarized in the domain submodel in Figure 3 and definitions for the terms

[16] https://github.com/openai/whisper, accessed 26.03.2024
[17] https://www.maxqda.com/de/produkte/maxqda-analytics-pro, accessed 26.03.2024

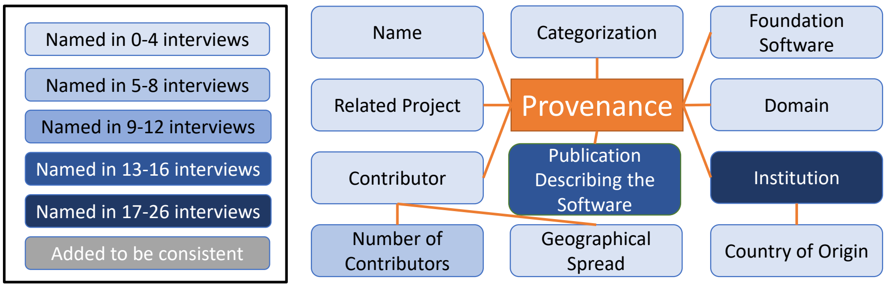

Figure 3: Detailed Domain Model for Provenance

Term	Definition
Categorization	Research category for which the software was written.
Contributor	A person who has contributed to the software.
Country of Origin	Country of the institution.
Domain	Research domain for which the software was written.
Geographical Spread	Geographical spread of the contributors.
Institution	Institution which developed the software.
Name	Name of the software.
Number of Contributors	Number of contributors.
Publication Describing the Software	An academic publication related to the software.
Related Project	Name of the project during which the software was developed.
Software Foundation	Software this software is based on.

Table 4: Definitions of Provenance Elements

are listed in Table 4.

For example, researchers are interested in general background information of the software, such as a publication. Also, they want to know how many people contributed to the software. The number is used as an indicator of the quality of the software by the researchers (I27: "So if it's like [...] there's a bunch of people involved in this community, it must be good.").

Regarding the institution behind the software, the interviewees can be divided into two groups. The first group is interested in this information for multiple reasons: if an institution with a high reputation is developing the software, that can be an indicator for a bigger user base (I9: "But maybe if there are researchers at MIT that developed something, they could do a better job in publicizing this and as a result of this, they have a larger user base.") or can be an indicator for quality if the researcher is familiar with the research group behind the software and the typical quality of that group's work. Information about the actor behind a software also allows the researchers to see possible economic interests behind the software. On the other side, some interviewees (a smaller portion) are not interested in the information, or even expect that they will have an unwanted bias towards software from institutions with better reputations when this information is given.

4.2 Usage

The category "usage" summarizes information that researchers want to know to simplify their usage of the software. This includes information on different versions, the given support, and future development of the software. All elements are displayed in Figure 4 and listed with definitions in Table 5.

Concerning versions, researchers would like to see information about how often the software is updated (I7:"Which I [...] think is a very important indicator [...] is how long, I say, or how often the code is updated.") and when the last version was published. Based on this information, they would like to get an impression that problems or bugs they find in the software will be fixed on short notice.

Researchers are especially interested in examples to see if they can directly use the software (I25: "One of the big things that I look for in software is are there examples that are provided with

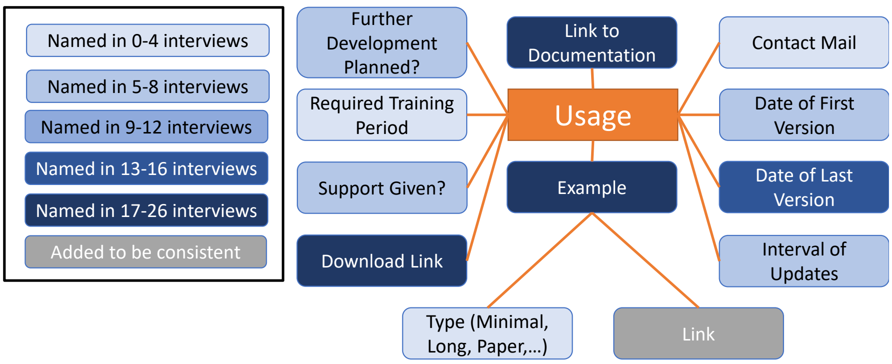

Figure 4: Detailed Domain Model for Usage

Term	Definition
Contact Mail	Individual responsible for maintaining the software.
Date of First Version	The date on which the CreativeWork was initially created.
Date of Last Version	The date on which the CreativeWork was most recently modified.
Download Link	If the file can be downloaded, URL to download the binary.
Example	Examples for using the software.
Example/Link	Link to the example.
Example/Type	Type of the example (e.g. minimal)
Further Development Planned?	Is a further development planned?
Interval of Updates	How often is the software updated?
Link to Documentation	Link to documentation.
Required Training Period	How long would a researcher need to be able to use the software?
Support Given?	Will support be given?

Table 5: Definitions of Usage Elements

the software"). They want to understand how to use the software and want to be able to directly run the software. Besides examples, they would also like to see a link to the documentation of the software.

4.3 Community

Within "community", different aspects of engagement with the community as well as estimating the size of the community are combined. Figure 5 gives an overview of all elements of this category while Table 6 lists the definitions for the elements. In general, researchers would like to know how many users a software has, e.g., by getting the number of downloads for the software.

Researchers are interested in knowing for which publications the software was already used.

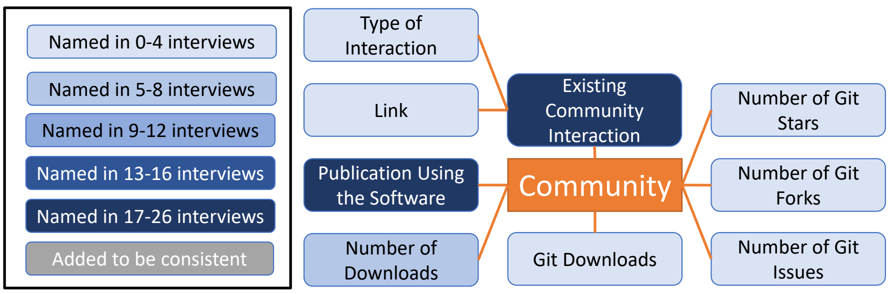

Figure 5: Detailed Domain Model for Community

Term	Definition
Existing Interactions	Ways how the software developers interact with the software users (community).
Existing Interactions/Link	Link to further information on the community interaction.
Existing Interactions/Type of Interaction	Type of the interaction.
Git Downloads	Number of Git Downloads.
Number of Downloads	Number of downloads.
Number of Git Forks	Number of forks on Git.
Number of Git Issues	Number of Git issues.
Number of Git Stare	Number of Git stars.
Publications Using the Software	A publication that uses the software for its research.

Table 6: Definitions of Community Elements

On one side, the pure number is interesting for the researchers (I7: ”The number of scientific papers that cite the software would actually be super cool.”) as it is a good equivalence to paper citations. On the other side, researchers are looking for examples of what can be done with the software and how it can be done. They expect to find this information in the publications using the software (I9: ”So, in many cases, you do not know how to initialize these parameters, and if you see that someone else has used this software in their research, you can also look for, hopefully, that particular section in the paper that explains how this is done, and you can reproduce this and be able to use that software.”).

Additionally, researchers would like to know if there is a community around the software and how it interacts. This interaction can for example include issues on GitHub or GitLab, mailing lists, or even community events. With this information, the researchers would like to estimate how easy or difficult it is to get support when using the software (I15: ”I think a mailing list [...] is very good, because it gives you the opportunity to share your problem with the community of users [...]”).

4.4 Interface

The category ”interface” summarizes all information needs around the interaction of an ERS with the outside world. This includes application programming interfaces (APIs), information on the type of input the software can read from files (e.g., specified via the command line.), or information on the type of output the software produces (e.g., into files or the command line). All information elements are displayed in Figure 6 and Table 7. Typically, software has multiple inputs and outputs. Also, some software can be integrated into other software or software allows the integration of other software components, e.g., certain models.

Researchers are interested in knowing the interface of the software including syntax, semantics, a general API description, and the information if the API follows a standard (I21: ”But also standardized interfaces to control laboratory devices when it comes to recording measurement

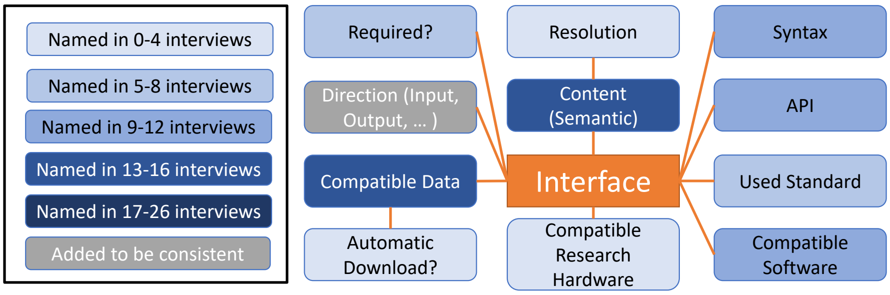

Figure 6: Detailed Domain Model for Interface

Term	Definition
API	Description of the input and output functions (APIs) of the software.
Compatible Research Data	Research data that is compatible with this software.
Compatible Research Data/Automatic Download	Is compatible data automatically downloaded by the software?
Compatible Research Hardware	Research hardware that is compatible with this software (e.g. in labs).
Compatible Software	A software which is compatible with this software.
Content (Semantic)	What type of data is usable as input?
Content (Semantic)/Resolution	Resolution of the variable.
Direction (Input, Output, ..)	How does the connected element relate to the software?
Required?	What data is required to get the software running?
Syntax	What should the data look like?
Used Standard	Standards for In/Output

Table 7: Definitions of Interface Elements

data. There are standards, so it would be good if they were at least partially supported.”). This information is useful to the researchers to roughly estimate how difficult it is to include the software in their ecosystem of other software and/or laboratory hardware.

Besides the description of the interfaces, researchers would like to have an overview of compatible data and software (I20: ”Not only compatible but what other software can be used together with it?”). Besides better estimating the required integration work, researchers also hope to find additional relevant software based on this information.

4.5 Software-specific

In the context of ”software-specific” information, some aspects are already top-level categories by themselves, e.g., usage and interface. The remaining software-specific information can further be separated into: quality, licenses, technical requirements, and performance. Figure 7 gives an overview of the identified information elements. Table 8 lists definitions for all elements.

In the context of performance, most researchers see a problem in simply comparing execution times since they highly depend on the used hardware. Nevertheless, an estimated execution time would already be helpful for the researchers. Also, they would like to know if the software can be run on a laptop or if a computing cluster is required (I25: ”Do I need to deploy this on an HPC cluster [...]?”).

Concerning the quality of research software, tests were often mentioned but are also controversial. Some researchers state that knowing the number of the tests without knowing the quality of the tests is not useful. (I20: ”That depends on how good the tests are. I think the number is not reliable.”). On the other hand, interviewees see having testing as already a quality indicator for research software (I26: ”That’s always a good indication that you’ve at least thought about a few

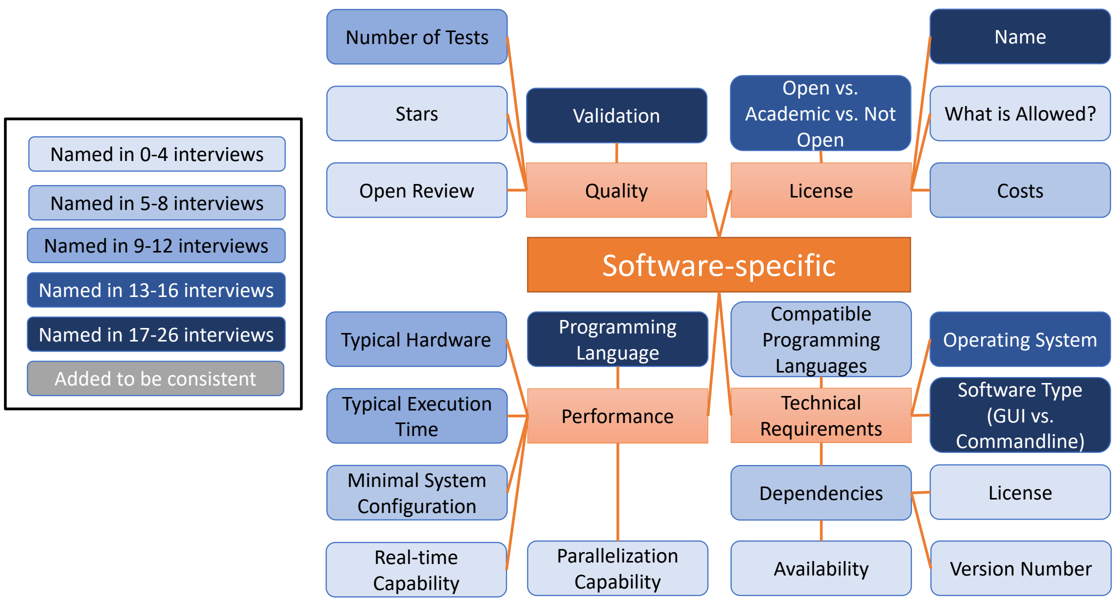

Figure 7: Detailed Domain Model for Software-Specific

Term	Definition
Compatible Programming Languages	Target programming environments to which the code applies. If applies to several versions, just the product name can be used.
Dependencies	Required software dependencies.
Dependencies/Availability	The availability of the dependency.
Dependencies/License	The license of the dependency.
Dependencies/Version Number	The version number of the dependency.
Licenses/Costs	Typical cost for licensing the software.
Licenses/Name	Name of the license.
Licenses/Open vs. Not Open vs. Academic	A flag to signal that the software is accessible for free.
Licenses/What is Allowed?	Details on the allowed reuse with these licenses.
Minimal System Configuration	Minimal hardware requirements.
Operating System	Operating systems supported (Windows 7, OSX 10.6, Android 1.6).
Parallelization Capability	Does the software support parallelization?
Programming Language	The computer programming language.
Quality/Number of Tests	Number of tests
Quality/Open Review	A review of the software.
Quality/Stars	Rating of the software in stars.
Quality/Validation	Information on validations done for the software.
Real-time Capability	Does the software support real-time simulations?
Software Type (GUI vs. Commandline)	How can I interact with the software?
Typical Execution Time	Typical execution time of the software.
Typical Hardware	Typical hardware which is required to run the software.

Table 8: Definitions of Software-specific Elements

tests. So that's already a plus point.").

For the technical requirements, researchers want to get information about which programming languages the software is compatible/usable with, especially if they have an engineering background (I3: "But if I have to integrate it into my programs myself, then it has to have a Python interface and yes, I think that's my criterion."). Also, they want to know the compatible operating systems, which sometimes present a clear requirement to a research software (I27: "I should say, I do have a preference if it doesn't work on a Mac. I don't have a Windows machine. And there's some energy software that doesn't work on Macs like Energy Plus or yeah, CAD software.").

For license information, some researchers only require general information about if it is open-source software while others want to know the exact license. In the context of commercial licenses, the researchers want to know possible license fees (I6: "I don't think I've mentioned license costs yet. That is of course also fundamental. Typically, of course, we don't have huge budgets to procure software licenses in the research projects.").

4.6 Functionalities

Within "functionalities", different information elements on the functionalities of ERS are combined. Figure 8 shows all elements of this category with Table 9 listing the definitions. Researchers want to get a fast overview of the relevant functionalities of the software to estimate if the software fits their use case and requirements. Therefore, many researchers want to see a graphical abstract of the software, e.g., an UML diagram of the primary functions (I11: "But then a small, for example, abstract or an UML diagram is always nice to see. Or simply a flow diagram, a black-box model, or small boxes that say this goes into a model and this comes out. Simply a small illustration that tries to explain the model to me in fewer words.").

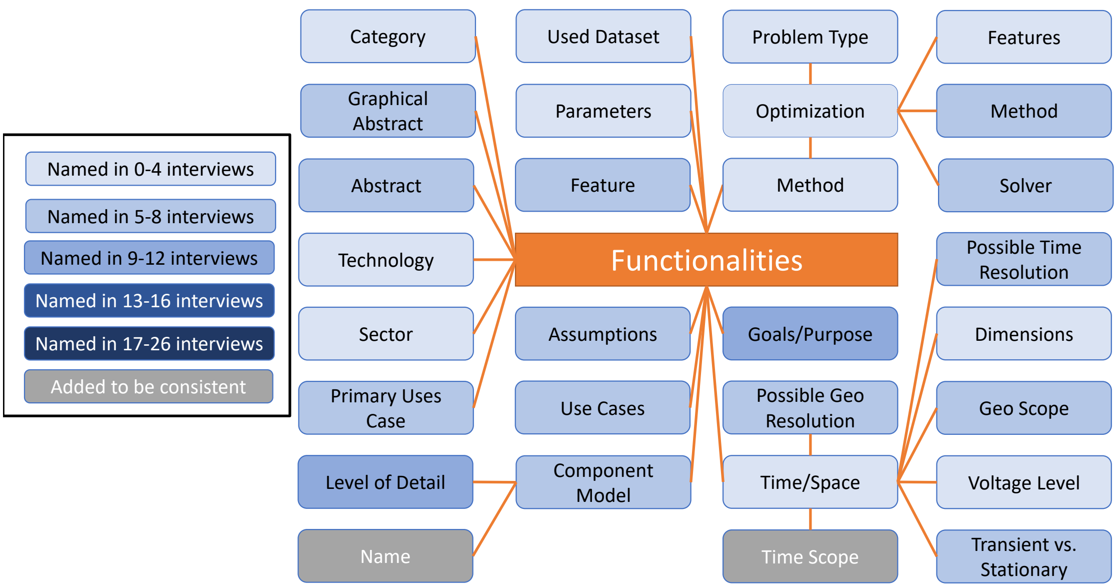

Figure 8: Detailed Domain Model for Functionalities

Term	Definition
Abstract	A description of the software.
Assumptions	A required assumption of the software.
Category	Research category for which the software was written.
Component Model	A component of the energy system included in the software.
Component Model/Level of Detail	Level of detail of the component.
Component Model/Name	Name of the component.
Feature	A feature of the software.
Goals/Purpose	A general goal of the software.
Graphical Abstract	Graphical overview on the functionality of the software.
Method	An used method in the software.
Optimization	Optimization used in the software.
Optimization/Features	Features of the optimization.
Optimization/Method	Method of the optimization.
Optimization/Problem Type	Problem class of the optimization.
Optimization/Solver	Used solver for the optimization.
Parameters	Relevant parameters of the software.
Primary Use Case	Primary use case of the software.
Sector	A supported sector of the software.
Technology	A supported technology of the software.
Time-Space	Time and space required for using the software.
Time-Space/Dimensions	Supported dimensions of the software.
Time-Space/Geo Scope	Support geographical scope of the software.
Time-Space/Possible Time Resolution	Supported time resolution of the software.
Time-Space/Transient vs. Stationary	Does the software support stationary or transient effects?
Time-Space/Voltage Level	Supported voltage level of the software.
Use Case	A supported use case.
Used Datasets	Is compatible data used within the software?

Table 9: Definitions of Functionalities Elements

To estimate if the software fits the problem, the researchers would like to see for which use-case the software was originally developed and which use-cases are currently supported by the software (I15: "Why was this software created? Which problem can now be solved with this tool, which was not reasonably solvable before?").

In the energy domain, geographic and time scales are important factors. Therefore, researchers want to know the geographical and time scope[18] of the software as well as the possible geographical and time resolutions. Based on this information, researchers can decide if the software fits their requirements (I18: "This also makes it easier to evaluate, whether this is a similar flight altitude to the one you are looking for."). When looking at certain components, the covered spatial dimensions of a model are also relevant. Concerning energy grids, the information if transient effects[19] are covered and which voltage levels[20] are included are also of interest (I17: "In thermal systems, we often do not know before we use libraries, whether transient behavior can be covered or whether capacitive behavior can be included.").

The researchers also want to know which sectors[21] and technologies are covered and modeled. They often look for models of specific components in energy systems that they need in their research (I1: "Are models available for pumps? Are models available for storage tanks?"). Also, the level of detail of the components is relevant. The models need to be detailed enough to include the relevant effects but should not be too detailed because that slows down the overall simulation (I18: "If this is now modeled in more detail, then it probably includes more physical effects, which run on time scales that are not relevant for us and would then make the simulation slower.").

To see if a software fits, researchers would also like to get a quick overview of the assumptions for the software. Since many software deals with optimization, researchers also want detailed information on the optimization, e.g., the used solver or the used method.

5 Discussion

As with all studies, our requirement analysis has some limitations that also need to be taken into consideration. Firstly, the analysis largely focused on requirements for metadata in a registry, specifically addressing the findability and selection of research software which is the main use case for research software metadata in the FAIR principles. However, other aspects, such as detailed provenance or software compatibility, were not included and require further investigation. Secondly, although efforts were made to incorporate international researchers, a significant number of participants were based in Germany (∼80%). This may have led to a narrower perspective on the requirements, and a more diverse pool of participants should be considered in future studies. Also, the analysis only focused on requirements from researchers and did not include requirements from other relevant stakeholders of the science system, such as editors or funding agencies. Their

[18] While some energy researchers cover large geographic areas like Europe, others only analyze the energy system of a single house.

[19] When analyzing electricity or thermal networks often transient effects are ignored to faster compute the networks. Depending on the research it is important to know if these effects are included or not.

[20] The electricity grid is categorized into different main voltage levels. Depending on the level the characteristics of the components as well as the relevance of the physical effects differs.

[21] Sectors refers on one hand to consumption sectors (e.g., households, industry, ..) and on the other hand to energy type sectors (e.g., heat, electricity)

input could provide valuable insights and contribute to a more comprehensive understanding of the requirements from different perspectives.

All in all, we showed the diverse information requirements of energy researchers when searching for ERS and when selecting an ERS to reuse for their purpose. The requirements contain domain-specific aspects of ERS but also show the need for many general information on research software which is not yet completely included in general approaches such as *CodeMeta*. One of these aspects is the topic of software quality, which multiple interviewees identified as important and difficult to measure. This is in line with the results of Eisty et al. [ETC18] whose analysis found that the knowledge of software metrics, and specifically of software quality metrics, is generally low with researchers developing software. Since this study only included energy researchers, it would be interesting to do more similar studies in other scientific disciplines.

The overview on the different metadata elements also shows the need for regularly updating the metadata, e.g., "Date of Last Version". This is a known problem for research software metadata [CKB+22]. Druskat et al. propose to directly update research software metadata in repositories and registries as part of CI pipelines [DBJ+22].

Overall, the idea to include ERS in a registry to make it more findable and reusable was evaluated positively in nearly all interviews. In respect to current search strategies for research software, the answers were highly diverse. Search engines, publications, and colleagues were mentioned the most times (by 8-9 participants each), while a few also mentioned software catalogs, conferences, and social media. Search engines were named less often than in the studies of Stevens (∼83%) [Ste22] and Hucka and Graham (∼91%) [HG18].

The developed domain model only focuses on the requirements for metadata for ERS. In the next step, the elements of existing metadata schemes for research software, especially the formalized general ones (see Table 1), need to be further analyzed concerning the requirements of energy researchers that were presented in this work. Based on this environmental scan, it can be decided which new metadata elements are required and which existing elements can be reused [CB13]. Existing elements should be reused as much as possible. Also, each element should be analyzed to determine if the values should be limited by using value vocabularies. Together with the presented analysis, a metadata scheme for ERS can be defined which will enable FAIR ERS. The scheme should be implemented for usage in the energy research community as part of the work of NFDI4Energy[22].

Acknowledgements: The authors would like to thank the anonymous reviewers for their valuable comments which helped to improve this paper. Also, the authors would like to thank Amanda Wein for her helpful proofreading of the manuscript.

The authors would like to thank the German Federal Government, the German State Governments, and the Joint Science Conference (GWK) for their funding and support as part of the NFDI4Energy consortium. The work was partly funded by the German Research Foundation (DFG) – 501865131 within the German National Research Data Infrastructure (NFDI, www.nfdi.de).

Additionally, this research was partly funded by the Lower Saxony Ministry of Science and

[22] NFDI (German National Research Data Infrastructure) is an initiative by the federal and state governments in Germany to develop and consolidate infrastructure for research data. NFDI4Energy is the consortium for interdisciplinary energy systems research.

Culture under grant number 11-76251-13-3/19–ZN3488 (ZLE) within the Lower Saxony "Vorab" of the Volkswagen Foundation. It was supported by the Center for Digital Innovations (ZDIN).

Bibliography

[BCK+22] M. Barker, N. P. Chue Hong, D. S. Katz, A.-L. Lamprecht, C. Martinez-Ortiz, F. Psomopoulos, J. Harrow, L. J. Castro, M. Gruenpeter, P. A. Martinez, T. Honeyman. Introducing the FAIR Principles for research software. *Scientific Data* 9(1):622, Oct. 2022. Number: 1 Publisher: Nature Publishing Group.
doi:10.1038/s41597-022-01710-x

[CB13] M. Curado Malta, A. A. Baptista. Me4DCAP V0.1: a Method for the Development of Dublin Core Application Profiles. *Mining the Digital Information Networks*, pp. 33–44, 2013.
doi:10.3233/978-1-61499-270-7-33

[CKB+22] 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, T. Honeyman, A. Struck, A. Lee, A. Loewe, B. van Werkhoven, C. Jones, D. Garijo, E. Plomp, F. Genova, H. Shanahan, J. Leng, M. Hellström, M. Sandström, M. Sinha, M. Kuzak, P. Herterich, Q. Zhang, S. Islam, S.-A. Sansone, T. Pollard, U. D. Atmojo, A. Williams, A. Czerniak, A. Niehues, A. C. Fouilloux, B. Desinghu, C. Goble, C. Richard, C. Gray, C. Erdmann, D. Nüst, D. Tartarini, E. Ranguelova, H. Anzt, I. Todorov, J. McNally, J. Moldon, J. Burnett, J. Garrido-Sánchez, K. Belhajjame, L. Sesink, L. Hwang, M. R. Tovani-Palone, M. D. Wilkinson, M. Servillat, M. Liffers, M. Fox, N. Miljković, N. Lynch, P. Martinez Lavanchy, S. Gesing, S. Stevens, S. Martinez Cuesta, S. Peroni, S. Soiland-Reyes, T. Bakker, T. Rabemanantsoa, V. Sochat, Y. Yehudi, R. F. WG. FAIR Principles for Research Software (FAIR4RS Principles). May 2022. Publisher: Zenodo.
doi:10.15497/RDA00068

[DBJ+22] S. Druskat, O. Bertuch, G. Juckeland, O. Knodel, T. Schlauch. Software publications with rich metadata: state of the art, automated workflows and HERMES concept. Jan. 2022. Gives an intersting list on relevant metadata.
http://arxiv.org/abs/2201.09015

[ETC18] N. U. Eisty, G. K. Thiruvathukal, J. C. Carver. A Survey of Software Metric Use in Research Software Development. In *2018 IEEE 14th International Conference on e-Science (e-Science)*. Pp. 212–222. Oct. 2018.
doi:10.1109/eScience.2018.00036
https://ieeexplore.ieee.org/abstract/document/8588655

[FN23] S. Ferenz, A. Nieße. Towards Improved Findability of Energy Research Software by Introducing a Metadata-based Registry. *ing.grid* 1(2), Nov. 2023. Number: 2

Publisher: Universitäts- und Landesbibliothek Darmstadt.
doi:10.48694/inggrid.3837

[GGMR16] Y. Gil, D. Garijo, S. Mishra, V. Ratnakar. OntoSoft: A distributed semantic registry for scientific software. In *2016 IEEE 12th International Conference on e-Science (e-Science)*. Pp. 331–336. Oct. 2016. ZSCC: 0000007.
doi:10.1109/eScience.2016.7870916

[GMB+22] M. Grönewald, P. Mund, M. Bodenbrenner, M. Fuhrmans, B. Heinrichs, M. S. Müller, P. F. Pelz, M. Politze, N. Preuß, R. H. Schmitt, T. Stäcker. Mit AIMS zu einem Metadatenmanagement 4.0: FAIRe Forschungsdaten benötigen interoperable Metadaten. 2022. Publisher: heiBOOKS.
doi:10.11588/HEIBOOKS.979.C13721

[GOK+19] 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)*. Pp. 349–358. Sept. 2019.
doi:10.1109/eScience.2019.00046

[GRG15] Y. Gil, V. Ratnakar, D. Garijo. OntoSoft: Capturing Scientific Software Metadata. In *Proceedings of the 8th International Conference on Knowledge Capture*. K-CAP 2015, pp. 1–4. Association for Computing Machinery, New York, NY, USA, Oct. 2015.
doi:10.1145/2815833.2816955

[GS17] B. Glaser, A. L. Strauss. *Discovery of Grounded Theory: Strategies for Qualitative Research*. Routledge, July 2017.

[HG18] M. Hucka, M. J. Graham. Software search is not a science, even among scientists: A survey of how scientists and engineers find software. *Journal of Systems and Software* 141:171–191, July 2018.
doi:10.1016/j.jss.2018.03.047

[IIR+21] J. Ison, H. Ienasescu, E. Rydza, P. Chmura, K. Rapacki, A. Gaignard, V. Schwämmle, J. Van Helden, M. Kalaš, H. Ménager. BiotoolsSchema: A formalized schema for bioinformatics software description. *GigaScience* 10(1), 2021.
doi:10.1093/gigascience/giaa157

[JBM+17] M. B. Jones, C. Boettiger, A. C. Mayes, P. Slaughter, Y. Gil, N. Chue Hong, C. Goble. CodeMeta. 2017. ZSCC: 0000000 Publisher: Github.
http://ssi1.eprints-hosting.org/id/eprint/2/

[KGH21] D. S. Katz, M. Gruenpeter, T. Honeyman. Taking a fresh look at FAIR for research software. *Patterns* 2(3), Mar. 2021.
doi:10.1016/j.patter.2021.100222

[MBL+14] J. Malone, A. Brown, A. L. Lister, J. Ison, D. Hull, H. Parkinson, R. Stevens. The Software Ontology (SWO): a resource for reproducibility in biomedical data analysis, curation and digital preservation. *Journal of Biomedical Semantics* 5(1):25, June 2014.
doi:10.1186/2041-1480-5-25

[MN07] M. D. Myers, M. Newman. The qualitative interview in IS research: Examining the craft. *Information and Organization* 17(1):2–26, Jan. 2007. ZSCC: 0002318.
doi:10.1016/j.infoandorg.2006.11.001

[NS15] H. Noble, J. Smith. Issues of validity and reliability in qualitative research. *Evidence-Based Nursing* 18(2):34–35, Apr. 2015. Publisher: Royal College of Nursing Section: Research made simple.
doi:10.1136/eb-2015-102054
https://ebn.bmj.com/content/18/2/34

[SC90] A. L. Strauss, J. M. Corbin. *Basics of qualitative research: Grounded theory procedures and techniques*. Basics of qualitative research: Grounded theory procedures and techniques. Sage Publications, Inc, Thousand Oaks, CA, US, 1990.

[Sil16] D. Silverman (ed.). *Qualitative research*. Sage, Los Angeles, 4e edition, 2016.

[SL19] J. Schwarz, S. Lehnhoff. Ontological Integration of Semantics and Domain Knowledge in Energy Scenario Co-simulation. In *Proceedings of the 11th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*. Pp. 127–136. SCITEPRESS - Science and Technology Publications, Vienna, Austria, 2019.
doi:10.5220/0008069801270136

[Ste22] F. Stevens. Understanding how researchers find research software for research practice. Technical report, Zenodo, Dec. 2022.
doi:10.5281/zenodo.7340034

[WDA+16] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L. B. d. S. Santos, P. E. Bourne, J. Bouwman, A. J. Brookes, T. Clark, M. Crosas, I. Dillo, O. Dumon, S. Edmunds, C. T. Evelo, R. Finkers, A. Gonzalez-Beltran, A. J. G. Gray, P. Groth, C. Goble, J. S. Grethe, J. Heringa, P. Hoen, R. Hooft, T. Kuhn, R. Kok, J. Kok, S. J. Lusher, M. E. Martone, A. Mons, A. L. Packer, B. Persson, P. Rocca-Serra, M. Roos, R. v. Schaik, S.-A. Sansone, E. Schultes, T. Sengstag, T. Slater, G. Strawn, M. A. Swertz, M. Thompson, J. v. d. Lei, E. v. Mulligen, J. Velterop, A. Waagmeester, P. Wittenburg, K. Wolstencroft, J. Zhao, B. Mons. The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data* 3(1):1–9, Mar. 2016. Number: 1 Publisher: Nature Publishing Group.
doi:10.1038/sdata.2016.18