Bausteine Forschungsdatenmanagement
Empfehlungen und Erfahrungsberichte für die Praxis von Forschungsdatenmanagerinnen und -managern

Connecting RDM Services

Interoperability and Interfaces for RDMO

Jürgen Windeck[i] Max Schröder[ii] Johannes Frenzel[iii]
Stefan Kusmierz[iv] Hannes Fuchs[v] Jochen Klar[vi]

2024

Zitiervorschlag

Windeck, Jürgen, Schröder, Max, Frenzel, Johannes, Kusmierz, Stefan, Fuchs, Hannes, Klar, Jochen. 2024. Connecting RDM Services - Interoperability and Interfaces for RDMO. *Bausteine Forschungsdatenmanagement. Empfehlungen und Erfahrungsberichte für die Praxis von Forschungsdatenmanagerinnen und -managern* Nr. 2/2024: S. 2-9. DOI: 10.17192/bfdm.2024.1.8629.

[i]ULB Darmstadt. ORCID: 0000-0003-1909-4353
[ii]Universitätsbibliothek Rostock. ORCID: 0000-0003-1522-494X
[iii]Ruhr Universität Bochum. ORCID: 0000-0003-0458-5424
[iv]Physikalisch-Technische Bundesanstalt, Braunschweig. ORCID: 0009-0006-4068-3418
[v]Helmholtz-Zentrum Potsdam Deutsches GeoForschungsZentrum GFZ. ORCID: 0009-0002-9257-0918
[vi]Freier Softwareentwickler. ORCID: 0000-0002-5883-4273

Abstract

The Research Data Management Organiser (RDMO) supports researchers in the creation of data management plans (DMPs). The open source software has established itself widely across Germany as the standard DMP tool. For its broader adoption, integration with other services for research data management is key. In this paper we reflect on the potential of RDMO on this issue, we explain its adaptiveness by design, its plugin architecture, and report on the first RDMO Hackathon organized in Darmstadt in Aug 2023 which focussed on the integration of RDMO via plugins. We present three extensions and plugins developed during the hackathon: 1.) a JSON-editor for RDMO data transfer and mapping, 2.) an RO-crate export plugin, 3.) a sensor database option set plugin.

1 Introduction

Since it was first released in 2016, the Research Data Management Organiser (RDMO) has established itself as the standard DMP tool in Germany with over 50 instances across universities, universities of applied sciences and other research institutions according to the RDMO website[1]. NFDI consortia using RDMO or are planning to[2], developing catalogs or trainings, are further demonstrating its standing in the German research data management services landscape. For the upcoming NFDI service architecture, discussed and developed mainly by the NFDI sections[3] and in the context of Base4NFDI[4] integration of services is of utmost importance. This raises the question on how RDMO supports this integration of RDM services.

To explore this question, we (1) introduce RDMO's architecture that enables the integration with other services and (2) demonstrate its potential by reporting on the results of the first RDMO Hackathon in 2023 at ULB Darmstadt that focused on respective plugins. As an open source software RDMO can be adapted to your own requirements, for which its built-in plugin architecture can be utilized to connect and integrate external resources.

2 RDMOs plugin architecture

Right from the start, the connection to external resources was an integral part of the development strategy of RDMO. Its application programming interface (API), which follows the representational state transfer paradigm (REST API) allows to read and

[1] https://rdmorganiser.github.io/. All websites in this article last retrieved Dec 22nd, 2023
[2] e.g. NFDI4Ing: https://rdmo.nfdi4ing.de
[3] Diepenbroek, Schimmler, & Ebert (2021)
[4] https://www.base4nfdi.de/

update the data of an RDMO project using scripting and external tools, and project content can be exported as JSON, CSV or using RDMO's own XML format. Recently, the API was enriched with additional features, e.g. with an additional endpoint to invite members to a project[5], allowing users to create their own personal API token for access[6], or to read the progress of a project[7].

Next to the API, RDMO includes a plugin architecture to create four different types of plugins for integration[8]:

1) *Export plugins* define a custom download format for project data. Plugins like the XML or CSV export are activated by default. Additionally, export plugins for DataCite and machine-actionable DMP (maDMP)[9] have been developed in the past[10]. An export plugin can not only create a file to be downloaded, but can also deposit it into a different infrastructure using the OAuth2 protocol for necessary authorization. An example for this is the creation of datasets in RADAR or Zenodo.

2) *Import Plugins* can be used to import data from different sources into a project. In the simplest case, this can be done via a file upload, but the file can also be fetched from a remote service (again including authentication).

3) With *option set providers* dynamic option sets can be defined in RDMO. Option sets are lists of items that are linked to a specific question in an RDMO questionnaire as checkboxes or radio buttons. In most cases, those option sets are defined during catalog development, but dynamic option sets are created during runtime based on a connection to an external API or database. For example, the re3data plugin[11] creates a list of all repositories listed in re3data, that can be implemented as a drop-down menu in the questionnaire. Similar plugins are available to query external vocabularies such as the Integrated Authority File (German: Gemeinsame Normdatei) (GND)[12] and WikiData[13].

4) *Issue providers* are able to connect RDMO to project management software. Tasks created in RDMO can be sent to services like OpenProject[14] as issues to be further processed. Integrations can also work both ways, e.g. webhooks can be used to trigger the closing of an issue in RDMO[15].

[5] https://github.com/rdmorganiser/rdmo/issues/540
[6] https://github.com/rdmorganiser/rdmo/issues/559
[7] https://github.com/rdmorganiser/rdmo/issues/429
[8] https://rdmo.readthedocs.io/en/latest/plugins/index.html
[9] On the topic of machine-actionable DMPs see, e.g. Miksa, T., Oblasser, S., & Rauber, A. (2022)
[10] https://github.com/rdmorganiser/rdmo-plugins
[11] https://github.com/rdmorganiser/rdmo-re3data
[12] https://www.dnb.de/DE/Professionell/Standardisierung/GND/gnd_node.html
[13] https://www.wikidata.org
[14] https://www.openproject.org
[15] https://github.com/rdmorganiser/rdmo-plugins

As these examples show, RDMO has demonstrated its potential to integrate with the existing RDM services landscape. It connects to registers or databases with metadata to be used in an RDMO questionnaire, it can trigger external services or provide integrations into project management solutions. RDMO is based on open formats and documentation and examples are available to create plugins and integrations customized to specific RDM solutions at institutions.

3 The RDMO Hackathon 2023

The idea for a meeting originated in a brief conversation about RDMO during the E-Science Tage 2023 in Heidelberg. Shortly thereafter, the decision was made to organize the first RDMO Hackathon during summer 2023. Fifteen participants from institutions all over Germany gathered at ULB Darmstadt to exchange ideas on possible RDMO plugins, extensions and develop together. One week before the actual live meeting, two online meetings were held to support newcomers in setting up their RDMO development environment on their computers, thus allowing for a smooth start when meeting in person. Pitches for plugins had been collected in the weeks before, making use of an online board allowing the whole RDMO community to post ideas. Altogether 16 ideas were collected[16] from which four topics were pitched in more detail at the beginning of the two-day meeting in Darmstadt. Three topics got sufficient support: 1.) a JSON editor for collecting and manipulating RDMO project data, 2.) an RO-Crate export plugin, 3.) a sensor database option set plugin. After a short basic introduction to the RDMO plugin architecture, three groups started their work. In the following, we describe the goals and results of each group.

3.1 A tool for metadata transfer and mapping - Metadata Ferry

As a tool for generating data management plans RDMO is the starting point of a chain of data management platforms participating in the research process. Electronic lab notebooks, research database organizers, open-access repositories and long-term repositories, to name but a few, demand almost without exception the input of metadata, which have already been entered into RDMO. To repeat this input alongside the research stream over and over again degenerates the work of a researcher to a stereotype, error-prone occupation.

Not least to remedy this ineffective and demotivating process, the development of a tool called “Metadata Ferry” that links the mentioned platforms and transfers the recorded data/metadata from RDMO to another platform, is desirable. The solution that Metadata Ferry represents is new and unique because it is tailor-made for the research data flow in use at Physikalisch-Technische Bundesanstalt (PTB) but on the

[16] https://miro.com/app/board/uXjVMF6CBeo=/

other hand is modular and therefore easily scalable. The prototype described here is a preliminary stage of the tool to be developed. During the design of the spec sheet for this project it became clear soon that part of the story will be the task of metadata mapping. And that - as a posterior step to the mapping – it will be preferable to adjust the metadata by two editors. JSON is considered the prioritized format, but it is good form to allow for other common formats such as XML and YAML. That is why the proposal to the hackathon consisted of the development of a prototype including two JSON editors, which could form the take-off board for the later evolution to a metadata and data mapping and transferring interchange platform with either an integrated or API-driven format converter.

The team opted for the implementation of a react package to integrate the editors into the prototype. This package looked the most handy and the one with the most practical features for editing and modifying metadata. After two days of work the wanted result showed up (see figure 1). The developed package is now implemented and running as a demonstrator on a linux server that has been acquired exclusively for this purpose at PTB. The prototype allows importing prepared project data from RDMO and manipulating the JSON file in both editors. Currently the tool which is designed as a web service enters its pilot phase. The scheduling for availability as a production version within PTB still has to be defined.

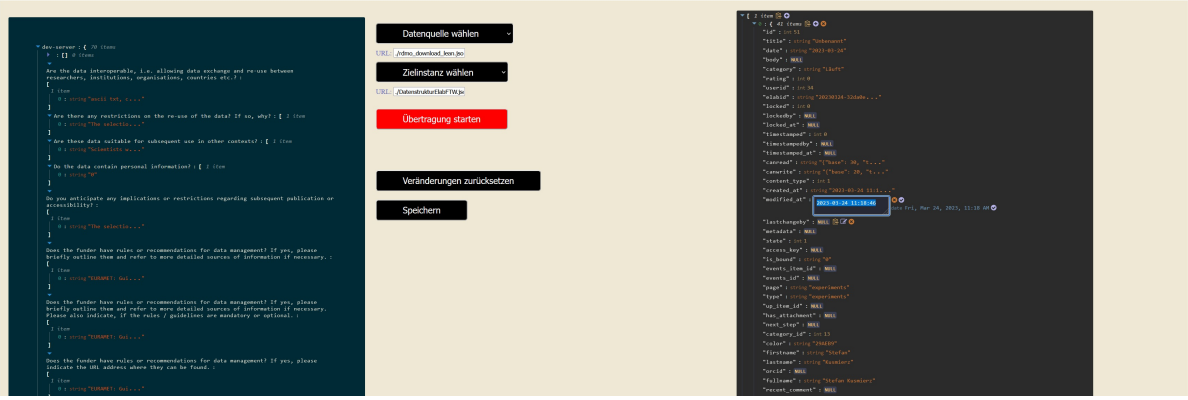

Fig. 1: Frontend of the Metadata Ferry prototype at the end of the hackathon: The two editors on the left and right show JSON files prospectively loaded via API from different RDM services. The content is to be manipulated in the editors and uploaded to the target system.

3.2 RO-Crate export plugin

The second group aimed to implement the RO-Crate[17] standard as an export format. The RO-Crate standard receives increasing attention, it forms the basis for the ELN

[17] https://www.researchobject.org/RO-crate/

exchange standard[18] and is also discussed as one potential implementation for the FAIR digital object[19]. An RO-Crate is a bundle of (1) a set of files and directories, and (2) its metadata in the form of semantic knowledge[20]. In particular, the metadata is encoded as JSON-LD by employing the RO-Crate specification. Thus, RO-Crates enable the machine-interpretable documentation of research data.

DMPs contain a lot of information with respect to the life cycle of research data. Thus, an RO-Crate export from within RDMO enables researchers to download the metadata collected in their DMP in a standardized structure. The resulting RO-Crate serves as a scaffold which can be completed by adding research data and additional metadata during the research process ideally by other tools used in that process, e.g. electronic lab notebooks. After the active research has been completed the crate eventually contains all necessary data and is ready to be archived or published in a data repository with no or only little extra curation effort for the researcher.

The implementation relies on the RO-Crate Python library[21]. When activated, the plugin is listed in the export section of the project overview, allowing the download of a zip container structured as an RO-Crate. The user can choose which datasets of his project should be exported. The motivation to allow for the selection of datasets is the increased flexibility by enabling the creation of separate RO-Crates for different datasets. Authors are linked to the datasets, if the catalog contains both entities.

In order to configure the mapping of RDMO project data to the RO-Crate metadata, a configuration file is used. During the hackathon a minimum set of metadata was employed to demonstrate its feasibility. A config file allows for easier adaptation, e.g. using a specific RDMO catalog that is making use of other domain attributes. After its initial implementation during the hackathon, a couple of adjustments, mostly code refactorings, had been made in the weeks following. The plugin is compatible with RDMO version 2.0 and can be downloaded via github[22].

3.3 Sensor option set plugin

This plugin demonstrates how questions in a catalog can be filled out with values from an external system. It is needed as a module to replace an existing system with RDMO in the MOSES context[23]. MOSES emphasizes the importance of instruments throughout the project lifecycle, starting with the planning phase. DMPs in MOSES serve as a project's guiding hand, encompassing all phases from planning to post-implementation. By meticulously documenting instruments, researchers can ensure informed planning, transparent data collection, and robust scientific publications.

[18] https://github.com/TheELNConsortium/TheELNFileFormat
[19] Castro, Soiland-Reyes, & Rebholz-Schuhmann (2023)
[20] Soiland-Reyes et al. (2022)
[21] https://github.com/researchobject/ro-crate-py
[22] https://github.com/rdmorganiser/rdmo-plugins-rocrate
[23] MOSES - Modular Observation for Earth Systems, https://www.ufz.de/moses/

The plugin makes use of the option set, which displays appropriate results depending on the user's input. As soon as the user selects and saves a result from the list (search field), additional information on the selected item is loaded and questions in the catalog are filled out automatically with this information. In this specific example, the API of a sensor information system is used[24] which is hosted by the Alfred-Wegener Institut[25]. The user can search for and select the desired sensor and information such as name, type and serial number are filled out without any actions required on the user's part. In the background, Django Signals[26] are used to send further requests to the API to obtain the relevant information and fill out the questions. This functionality will simplify the entry of information that is already available in other systems. The aim is to make work easier for users by automatically transferring information from other systems without the need for manual copy. Furthermore, incorrect entries, which can occur with manual input, can be avoided.

The next step is to extend the plugin so that it can communicate with other sensor APIs at the same time, thus enabling search and selection across multiple systems. It is also intended to read significantly more attributes and to add a configuration file. While this specific functionality is currently tailored for MOSES, the plugin can serve as a blueprint for simplifying user input from other systems through automated data filling. The ability to generate various export formats in RDMO opens up the possibility of reusing the information in other systems, fostering data integration and cross-platform collaboration.

4 Summary

The hackathon officially ended almost two months after the meeting in Darmstadt, giving each group time and space to work out the initial results. A final online meeting where open issues and ongoing work were discussed took place. The hackathon made it possible for both new and established RDMO developers to connect and exchange ideas on the topic of RDMO integration as well as the general development of RDMO in an effective way. The overall feedback unanimously advocated a Hackathon 2024.

As shown by these examples, RDMO is well prepared for the upcoming NFDI services landscape. The API and plugin architecture allow for different forms of service integration and the use of standards in RDMO. This enables institutions to integrate and extend the tool without the need to understand every detail of the source code. We expect numerous extensions for RDMO, connecting it to, e.g. vocabulary services, repositories, and research information systems. In the future, the plugin system will be

[24] AWI O2A: Data flow from Observations to Analysis & Archives - Web Services: https://spaces.awi.de/pages/viewpage.action?pageId=311778808
[25] https://www.awi.de/en/
[26] https://docs.djangoproject.com/en/4.2/ref/signals/

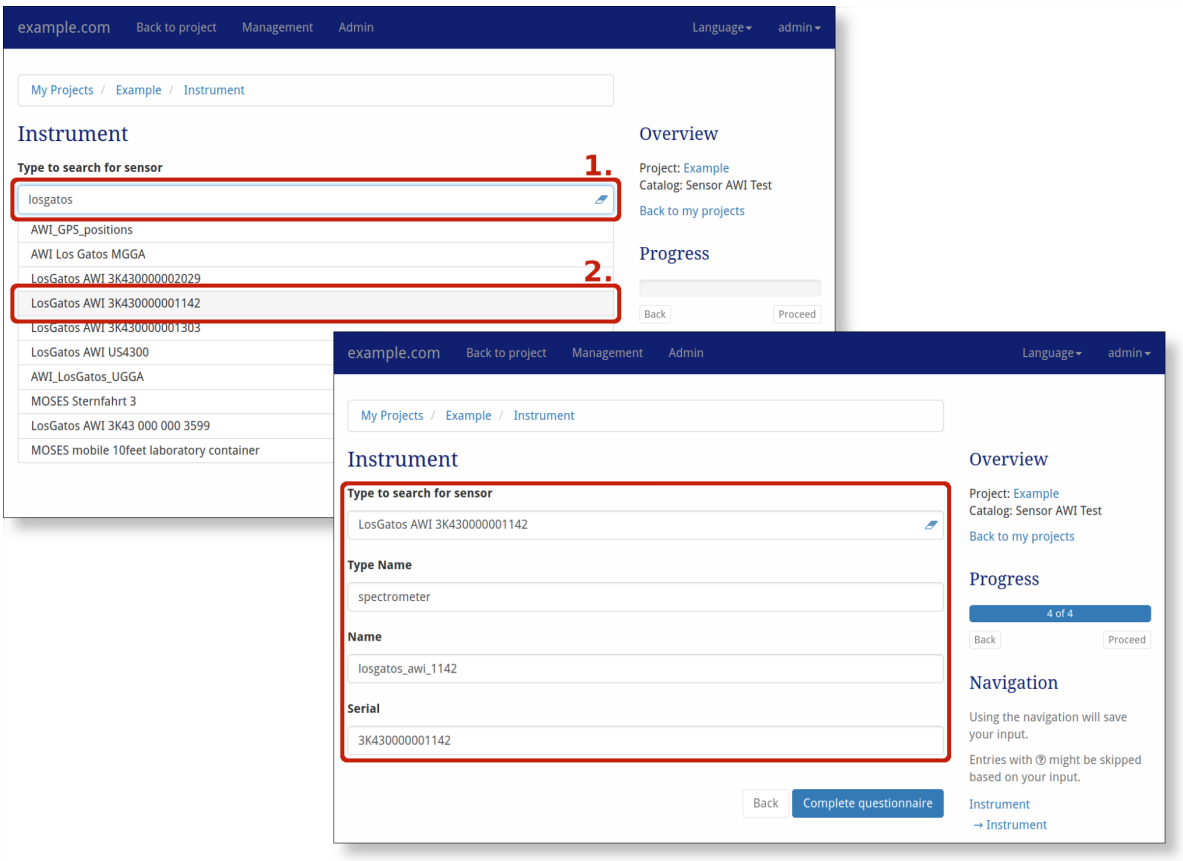

Fig. 2: Screenshots of the plugin workflow during the RDMO questionnaire: 1. User enters a search term, 2. Selects item from the list generated from querying the database via the search term. 3. Finally the fields are automatically filled out after selecting the item.

further developed to extend the output functionalities of RDMO to create more customizable templates to better fulfill the formal requirements from funding agencies and similar stakeholders. The internal vocabulary connects all content items and can be mapped or extended easily to domain-specific needs. The RDMO community is open for contributions.

5 Acknowledgements

The RDMO Hackathon was made possible by HeFDI (https://www.hefdi.de), the Hessian Research Data Infrastructures project, funded by the Hessian Ministry of Higher Education, Research and the Arts.

6 References

Castro, L.J., Soiland-Reyes, S., & Rebholz-Schuhmann, D. (2023). RO-Crates meets FAIR Digital Objects. https://doi.org/10.52825/cordi.v1i.396

Diepenbroek, M., Schimmler, S., & Ebert, B. (2021). Sektionskonzept Common Infrastructures zur Einrichtung einer Sektion im Verein Nationale Forschungsdateninfrastruktur (NFDI) e.V. Zenodo. https://doi.org/10.5281/zenodo.5607490

Miksa, T., Oblasser, S., & Rauber, A. (2022). Automating Research Data Management Using Machine-Actionable Data Management Plans. ACM Transactions on Management Information Systems, 13(2), 1–22. https://doi.org/10.1145/3490396

Soiland-Reyes, S., Sefton, P., Crosas, M., Castro, L. J., Coppens, F., Fernández, J. M., et al. (2022). Packaging research artefacts with RO-Crate. Data Science, 5(2), 97–138. https://doi.org/10.3233/DS-210053