SM4RO-C: SciMesh for RO-Crate

Release 1.0.0

Torsten Bronger **Michael Flemming**
Hartmut Schlenz **Michael Selzer**
Manideep Jayavarapu

Dec 08, 2022

Contents:

Date 2022-12-05

Abstract SciMesh is a set of specifications that define the representation of scientific results in form of a knowledge graph. RO-Crate is a container format to hold scientific data. In this paper, we present a way to combine both to create self-contained digital artefacts of scientific output that can be published, archived, and used to interchange data between scientific databases and electronic lab notebooks. We call it "SM4RO-C", pronounced "smaro C".

Authors

- Torsten Bronger[1], t.bronger@fz-juelich.de
- Michael Flemming[1], m.flemming@fz-juelich.de

[1] Forschungszentrum Jülich, ZB, Jülich, Germany

- Hartmut Schlenz[2], h.schlenz@fz-juelich.de
- Michael Selzer[3], michael.selzer@kit.edu
- Manideep Jayavarapu[3], manideep.jayavarapu@kit.edu

1 Motivation

Sharing scientific results in a machine-actionable manner is a big challenge. The plethora of possible kinds of results and insights and their complex interconnections makes it virtually impossible to cover all use cases. However, we do think that it is possible to provide re-usable scientific data for many cases. Moreover, one can deploy an approach that is extensible in a way that more and more research can be expressed over time, striving for almost-complete coverage.

We consider the graph as a suitable data structure for this endeavour. While not very efficient regarding the operations that act upon it, it can be extended arbitrarily, and extensions do not affect systems that were designed to deal with the non-extended graph. In other words, the producer can add all nodes (information) they can think of to the graph, while the consumer only processes the subgraph that they can understand.

We have already proposed [SciMesh] as a schema for RDF graphs that represent scientific workflows not limited to computations – and the results stemming from them. However, SciMesh deliberately does not address two aspects:

1. Quite often, data needs to have a well-defined container in order to enhance interoperability. SciMesh, in contrast, can be stored in a triple store, in serialised form (Turtle, XML, JSON-LD) on disk or as byte stream over a network, or as a special data structure in memory (e.g. using [RDFlib]). Its concrete representation is not part of SciMesh's specification.
2. For effective re-use, access to all interconnected data of all stages of the scientific workflow is needed. This includes so-called raw data. Indeed many nodes in a SciMesh graph point to bulk data, e.g. images or CSV tables. However, SciMesh does not impose any restrictions on those links. For instance, they can be HTTP URLs, links into the [IPFS], or free-form location descriptions. Furthermore, it is not guaranteed that the links are not broken or behind an access restriction. And finally, even the SciMesh graph itself might be incomplete because party of it are stored in a different location.

In the following, we propose a way to embed SciMesh graphs into a container to produce self-contained artefacts of scientific results and insights, together with their context, provenance, and bulk data.

[2] Forschungszentrum Jülich, IEK-1, Jülich, Germany
[3] Karlsruhe Institute of Technology KIT, Karlsruhe, Germany

2 Terminology

In this paper, we use the following definitions.

Bulk data Data that is not stored in a graph. The most important reason for that is that it cannot be sensibly stored in an RDF literal, or represented as an RDF subgraph. Quite often such data is called "raw data" but this term might be misleading, as also processed data can be bulk data. Image files, CSV tables, ZIP files etc. are typical examples of bulk data.

Crate This is an [RO-Crate].

ELN consortium This denotes [TheELNConsortium].

Graph data Data stored as triples in a graph.

Metadata We do not use this term in this paper. There are so many contradicting definitions out there that the term is difficult to use when you need precision.

3 Basic Concept

The starting point is the RO-Crate profile proposed by the ELN consortium, see its specification. It is a ZIP file containing the bulk data organised as directories and files, and a top-level file `ro-crate-metadata.json` that describes the presented research in general and the files themselves in particular, using [JSON-LD] for the syntax and [schema.org] for the vocabulary.

We will give some details of the file format in the following, but we must refer to the documentation of the ELN consortium and RO-Crate for the details.

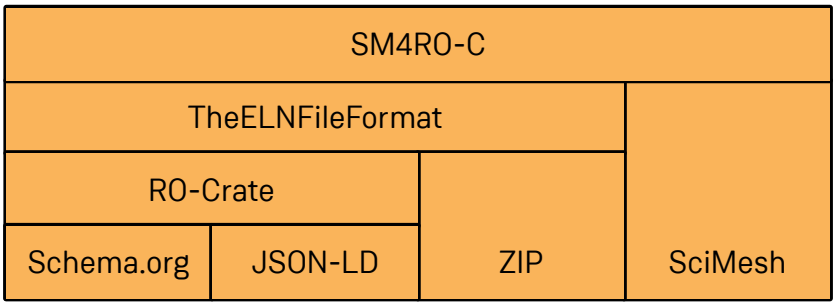

Fig. 3.1: Technology stack for SciMesh for RO-Crate.

The decisive addition of SM4RO-C to the file format of the ELN consortium is one or more `mainEntity` entries to the top-level `Dataset`. They point to top-level SciMesh entities, e.g. "Sample" or "Insight", which in turn are the starting points of full-fledged SciMesh graphs.

Technically, SM4RO-C is then a new profile of RO-Crate.

4 About Granularity

What should one create contain? It might be all the data of one sample. It might also be a sample series, or the data of a whole PhD thesis. Or, it might be only the data of one single experiment.

If you pack too much in a crate, it might be difficult to handle due to its size. Moreover, since it is a ZIP file, everything in it is opaque to the outside world. It can only be referred to as a whole. While theoretically technologies like [IPLD] could walk through the create, so that pointers to objects within the crate were possible, this is not implemented in practice and would mean an enormous latency.

If you pack too little in a crate (e.g. only one experiement), it may not be self-contained and of insufficient use for another scientist.

In the SM4RO-C implementation in [JuliaBase], each crate contains exactly one sample. This is considered a sensible level of granularity for exchanging data between ELNs, which is the primary purpose for the crate export. But if you want to use a crate for a data publication, it may be necessary to put more into one crate.

5 Anatomy of a SM4RO-C file

Remember that everything in the following is packed into a ZIP file.

At the top level, there is only one directory named like the crate (i.e. the ZIP file) itself. This directory contains all the bulk data organised in subdirectories and files. Furthermore, it must contain a file `ro-crate-metadata.json` with the following JSON:

```
{"@context": {
   "@vocab": "https://w3id.org/ro/crate/1.1/context",
   "sm": "http://scimesh.org/SciMesh/"
 },
 "@graph": [
   {
     "@type": "CreativeWork",
     "@id": "ro-crate-metadata.json",
     "conformsTo": {"@id": "https://w3id.org/ro/crate/1.1"},
     "about": {"@id": "./"}
   },
   {
     "@id": "./",
     "@type": "Dataset",
     "..."
   }
 ]
}
```

Note the two `"./"`: they connect the self-describing block of `ro-crate-metadata.json` with its root data entity. Both of these are the only mandatory elements in the `@graph`

array. All keys not starting with @ come from the context. You can visit the URL https://w3id.org/ro/crate/1.1/context and have a look at RO-Crate's vocabulary. Most of it is taken from [schema.org]. Please have a look at the documentation of [RO-Crate] or [JSON-LD] for how to add own vocabulary. However, because Schema.org dominates Crates, SciMesh tries hard to use it whenever feasible.

Now let's have a closer look at the root data entity, in particular what is behind the `"..."`.

5.1 Root data entity

The root data entity may just list the directories and files in the ZIP file:


```
{
  "@id": "./",
  "@type": "Dataset",
  "hasPart": [
    "./conductivity-setup-2",
    "./conductivity-setup-2/run-2022-12-01-321.csv",
    "./conductivity-setup-2/run-2022-12-03-322.csv",
    "./conductivity-setup-2/run-2022-12-03-323.csv",
    "./conductivity-setup-2/run-2022-12-03-324.csv",
    "./conductivity-setup-2/run-2022-12-04-325.csv",
    "./rem",
    "./rem/7243868563.png",
    "./rem/4237863643.png",
    "./rem/1325263347.png",
  ],
  "mainEntity": "https://eln.institute.example.com/samples/34"
}
```


After that, all the SciMesh graph nodes follow, for example the sample node:


```
{
  "@id": "https://eln.institute.example.com/samples/34",
  "@label": "14S-005",
  "@type": [
    "http://inm.example.com/Sample",
    "sm:Sample"
  ],
  "http://inm.example.com/Sample/currentLocation": "Rosalee's□
↪office",
  "http://inm.example.com/Sample/externalGraphUrls": "[]",
  "http://inm.example.com/Sample/lastModified": {
    "@type": "xmls:dateTime",
    "@value": "2022-12-08T10:11:04.090213+00:00"
  },
  "http://inm.example.com/Sample/purpose": "",
  "http://inm.example.com/Sample/tags": "",
  "jb-s:currentlyResponsiblePerson": {
```


(continues on next page)

(continued from previous page)


```
    "@id": "http://inm.example.com/User/7"
  },
  "jb-s:topic": "Cooperation with Paris University",
  "sm:state": {
    "@id": "http://inm.example.com/5-chamber_depositions/14S-005
↪#sample-5"
  }
}
```


Or, a process that was made with that sample:


```
{
  "@id": "http://inm.example.com/5-chamber_depositions/14S-005",
  "@label": "5-chamber deposition 14S-005",
  "@type": [
    "sm:Process",
    "http://inm.example.com/FiveChamberDeposition"
  ],
  "http://inm.example.com/Deposition/number": "14S-005",
  "http://inm.example.com/Deposition/splitDone": false,
  "jb-p:comments": "",
  "jb-p:finished": true,
  "jb-p:last_modified": {
    "@type": "xmls:dateTime",
    "@value": "2022-12-08T10:11:04.068996+00:00"
  },
  "jb-p:timestamp_inaccuracy": 0,
  "sm:cause": {
    "@list": []
  },
  "sm:operator": {
    "@id": "http://inm.example.com/User/7"
  },
  "sm:timestamp": {
    "@id": "_:n63dbe68802f346f195367f9b83b52a84b13"
  }
}
```


(Note that due to the handling of multi-sample processes in SciMesh, the sample points not directly to that process, although it is the latest one.)

5.2 Links to directories and files

The usual way to list and describe files in an RO-Crate holds also in SM4RO-C crates. However, *additionally* the SciMesh graph will contain local links to the files. It is possible – and probable – that the SciMesh nodes point to both the files contained in the crate, and to the original locations of the bulk data.

References

[SciMesh] Torsten Bronger, Michael Flemming, Hartmut Schlenz, Michael Selzer, Manideep Jayavarapu: *SciMesh*, 2022, https://scimesh.org

[RDFlib] RDFLib Team: *RDFlib*, https://rdflib.readthedocs.io

[IPFS] Juan Batiz-Benet: *IPFS – Content Addressed, Versioned, P2P File System*, 2014, arXiv:1407.3561

[RO-Crate] Stian Soiland-Reyes, Peter Sefton, Mercè Crosas, Leyla Jael Castro, Frederik Coppens, José M. Fernández, Daniel Garijo, Björn Grüning, Marco La Rosa, Simone Leo, Eoghan Ó Carragáin, Marc Portier, Ana Trisovic, RO-Crate Community, Paul Groth, Carole Goble: *Packaging research artefacts with RO-Crate*, Data Science 5(2), 2022, https://doi.org/10.3233/DS-210053

[TheELNConsortium] Nicolas Carpi et. al.: *The ELN Consortium*, https://github.com/TheELNConsortium

[JSON-LD] Manu Sporny, Dave Longley, Gregg Kellogg, Markus Lanthaler, Niklas Lindström: *JSON-LD 1.1*, 2020, https://www.w3.org/TR/json-ld/

[schema.org] Dan Brickley et. al.: *Schema.org*, https://www.w3.org/community/schemaorg/

[IPLD] Juan Batiz-Benet et. al.: *IPLD – Interplanetary Linked Data*, https://ipld.io/docs/

[JuliaBase] Torsten Bronger: *The samples database framework JuliaBase*, 2021, https://juliabase.org