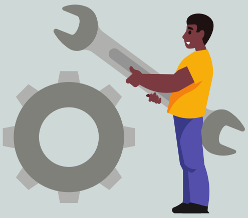
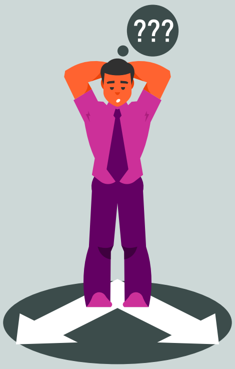


Tooling

FAIR Research Objects and Computational Workflows

Linked Data

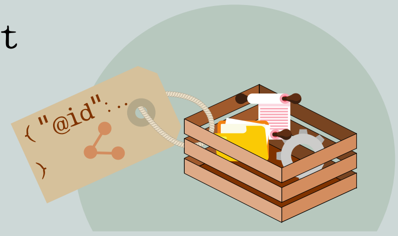
A Linked Data approach



Signposting

Reproducible

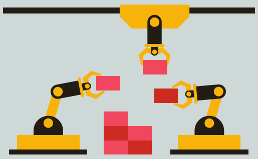
FAIR Digital Object



Research Object Crate

Workflow

Checklists



Machine actionable

Stian Soiland-Reyes

Interoperable

Accessible

Findable

Reusable

FAIR guiding principles

INDE lab
MANCHESTER
1824
eScience Lab

See Appendix B for full list of contributors, and Appendix A for funding acknowledgements.

Typeset in *TeX Gyre Pagella* using LuaLaTeX LuaHBTeX, Version 1.15.0 (TeX Live 2022)

Cover by Stian Soiland-Reyes, clip-art from https://publicdomainvectors.org/

Web: https://s11.no/2023/phd/
RO-Crate: https://w3id.org/ro/doi/10.5281/zenodo.8113625
DOI: https://doi.org/10.5281/zenodo.8113625
ISBN: 978-94-93391-78-9

FAIR Research Objects and Computational Workflows – A Linked Data Approach

Academisch Proefschrift

ter verkrijging van de graad van doctor
aan de Universiteit van Amsterdam
op gezag van de Rector Magnificus
prof. dr. ir. P.P.C.C. Verbeek

ten overstaan van een door het College voor Promoties ingestelde commissie,
in het openbaar te verdedigen in de Agnietenkapel
op woensdag 15 januari 2025, te 16.00 uur
door

Stian Soiland-Reyes

geboren te Stavanger

Promotiecommissie

Promoteres:		
	prof. dr. P.T. Groth	Universiteit van Amsterdam
	prof. dr. C.A. Goble	The University of Manchester
Overige leden:		
	prof. dr. R.V. van Nieuwpoort	Universiteit Leiden
	prof. dr. ir. C.T.A.M. de Laat	Universiteit van Amsterdam
	prof. dr. B.A. Plale	Indiana University Bloomington
	dr. Z. Zhao	Universiteit van Amsterdam
	dr. V.O. Degeler	Universiteit van Amsterdam

Faculteit der Natuurwetenschappen, Wiskunde en Informatica

Contents

List of Figures

List of Tables

List of Listings

Glossary

Application Programming Interface (API) Documented exposed set of methods and resources for programmatic use by a separate component, either within a programming language (e.g. Java interface declarations) or across a network protocol (e.g. REST services). 47, 72, 124, *see* REST

Comma-Seperated Values (CSV) Text-based format for representing two-dimensional data, rows are separated by newlines and fields by comma or semicolon. The format is very common (e.g. exported from Microsoft Excel spreadsheets) as it is relatively easy to consume or edit, however it is notoriously underspecified with many incompatible dialects and conventions. 77, 142, 148, 149, 193

Common Workflow Language (CWL) YAML-based file format for computational workflows, executable by a range of workflow engines [Crusoe 2022]. https://www.commonwl.org/ 123, 127, 133, 137, 138, 153, 157, 167, 171, *see* YAML

content-negotiation HTTP mechanism to request a desired media type of a resource that has multiple serialisation formats, e.g. `Accept: application/ld+json` to ask for JSON-LD. Commonly used on the Semantic Web for providing both a human-readable HTML representation and various RDF formats. 47, 189, 190, *see* HTTP, media type, JSON-LD & RDF

Create, Read, Update, Delete (CRUD) Basic data operations, e.g. for files, database rows, Web resources. 18, 39, 42, 49, 72

Digital Object Identifier (DOI) Handle-based persistent identifier, e.g. `10.5281/zenodo.8113625` — typically assigned to scholarly outputs (journal articles, datasets, reports) through one of the DOI registries (e.g. CrossRef, DataCite). Frequently expressed as an URI using the https://doi.org/ resolver. 16, 23, 138, 145, 194, *see* Handle, PID & URI

Digital Object Interface Protocol (DOIP) Text-based network protocol that exposes an API for a digital object, including basic CRUD operations, but also additional operations [DONA 2018]. The digital objects may optionally be addressed by Handle identifiers. 16, 19, 21, 33, 36, 38, 51, 52, 72, 150, *see* API, CRUD & Handle

Distributed System of Scientific Collections (DiSSCo) Research Infrastructure for natural science collections. https://www.dissco.eu/ 134, 139, 145, 151

EOSC Interoperability Framework (EOSC-IF) A set of recommendations for interoperability across services and data in the European Open Science Cloud [Kurowski 2021, Åkerström 2024]. See Section 3.1.3.6. 63, 190, *see* EOSC

European Open Science Cloud (EOSC) EU initiative to foster Open Science by integrating research infrastructures, interoperability standards, core services (e.g. persistent identifiers, authentication) and best practices [Ayris 2016] 31, 63, 71, 186, 187

FAIR Digital Object (FDO) Digital resource which follows certain guidelines and specifications to be machine-actionable and self-described, with an emphasis on identifiers, types, attributes and operations. See Section 2.1. 6, 9, 30, 31, 52, 71, 109, 121, 133–135, 138, 142, 148, 186, 187, 190, 207

Findable, Accessible, Interoperable, Reusable (FAIR) A set of principles for sharing of research data and metadata using machine-readable formats [Wilkinson 2016]. See Section 1.1.1. 2, 4, 6, 9, 17, 31, 39, 52, 72, 77, 109, 121, 133, 177, 186, 207

Handle The Handle system [Sun 2003a, Sun 2003b] is a distributed identifier system for digital objects. Each identified object (handle) (e.g. `2117/384589`) within a *prefix* has a set of key/value strings. Pre-defined keys `URL` `0.LOC` enable data retrieval from a server, additional keys can be added. The Handle protocol is also used for creating/updating handles. Expressed as URIs with the https://hdl.handle.net/ resolver, replacing the URN scheme `info:hdl:` 36, 37, 39, 145, 190

HyperText Markup Language (HTML) Text-based format used for representing Web page content, typically served over HTTP, styled with Cascading Style Sheets (CSS) and made interactive with JavaScript (JS). 23, 82, 99, 194, *see* HTTP

Hypertext Transfer Protocol (HTTP) Network protocol used for retrieving Web pages and invoking Web APIs [Fielding 1999]. The secure version of the protocol `https` adds encryption. 23, 37, 51, 72, 150, 187, *see* SSL & TLS

International Resource Identifier (IRI) Globally unique identifier string, equivalent to URIs, but permitting all international Unicode characters without needing escaping. Used as identifiers in JSON-LD. 23, 80, 81, 117, *see* URI & JSON-LD

Internet Protocol (IP) Routable network protocols used for all traffic on the Internet. IPv4 (e.g. address `8.8.8.8`) is gradually being replaced by IPv6 (e.g. `2001:4860:4860::8844`). 37

JavaScript Object Notation (JSON) A structured general purpose data format derived from the JavaScript programming language, commonly preferred by Web APIs. A JSON document consists of key-based objects (dictionaries), arrays, strings and numbers [Bray 2017]. 19, 49, 52, 79, 139, 142, 148, 150, 188

JSON Linked Data (JSON-LD) A JSON-based serialization of Linked Data [Sporny 2020]. Typically includes a `@context` which defines the mapping to RDF. Examples shown in Figure 2.2 and Listing 4.2. 19, 24, 27, 35, 39, 49, 72, 77, 81, 87, 117, 191, 202, 203, *see* JSON, LD & RDF

Linked Data (LD) Set of practices for publishing and relating data on the Web using RDF technologies [Bizer 2009]. See Section 2.2.2. 2, 5, 26, 30, 71, 72, 81, 127, 186, 188, 189, 207

Linked Data Platform (LDP) CRUD-like REST API for managing Linked Data containers of individual RDF resources [Sporny 2014] 38, 39, 52, *see* , REST, RDF & CRUD

media type Syntactic format of a bytestream within a network message, identified by a string. For instance, the HTTP header `Content-Type: text/html` in indicates the media type for HTML. There is an official Media Type registry[1]. 35, 47, 49, *see* HTML & HTTP

open Digital Specimens (openDS) Specification for describing digital twins of physical specimens [openDS 2021]. 139, 142, 148

optical character recognition (OCR) Computer method for generating digital text from an image, e.g. a scanned document or photograph of a label. 135, 136, 149

Persistent Identifier (PID) An identifier string that is globally unique, resolvable (typically through a resolver or registry) and with an organisational promise of persistence, suitable for inclusion in long-term archives and data publications [McMurry 2017]. 16, 18, 23, 37, 42, 72, 80–82, 138, 140, 148, 190, 196, 197, *see* URI & Handle

RDF Schema (RDFS) RDFS is a lightweight RDF vocabulary to define classes and properties for use in other RDF resources [Guha 2014]. 35, 49, 85, 194, 195

Representational State Transfer (REST) Architectural style for Web applications, where the stateless nature of Web is exploited by navigating the application state through Web resources, facilitating hypermedia formats [Fielding 2000]. *RESTful* Web Services exchanging JSON are the dominant form of Web APIs, although the hypermedia aspect is frequently neglected. 72, 124, *see* API, HTTP & JSON

Research Data Alliance (RDA) Community-led global initiative that build consensus on methods for open sharing and re-use of data and metadata. https://rd-alliance.org/ 7, 52, 197

[1] https://www.iana.org/assignments/media-types

Research Object (RO) Grouping of digital scholarly outputs with relationships, semantic descriptions and executable code [Bechhofer 2013]. See Section 1.2.2. 2, 8, 9, 77, 153, 194, 207

Research Object Crate (RO-Crate) Method to package Research Objects with metadata in an embedded JSON-LD file [Soiland-Reyes 2022a]. See Section 4.1. 76, 77, 128, 133, 152, 188, 203, *see* JSON-LD & RO

Research Software (RS) *Research Software includes source code files, algorithms, scripts, computational workflows and executables that were created during the research process or for a research purpose* [Gruenpeter 2021]. 5–7, 198, 206, 207

Resource Description Framework (RDF) Conceptual model for knowledge graph representation in the form of triples that relate Web resources [Schreiber 2014], realised through multiple serialisation formats. See Section 2.2.1. 2, 52, 71, 72, 81, 186, 188

Resource Description Framework in Attributes (RDFa) Mechanism to embed RDF statements within HTML documents by assigning attributes to elements, e.g. `<body vocab="http://purl.org/dc/terms/">` [Sporny 2015] 35, 117, *see* RDF & HTML

scientific workflow Data-driven computational workflow to *pipeline* scientific data for a particular analysis, e.g. data cleaning, image conversion. 7, 8, 63, 198

Secure Socket Layer (SSL) Cryptographic protocol commonly used on top of TCP/IP (e.g. `https://` URLs indicate HTTP over SSL). The secured connection is transparant - the application can treat the channel like any other. SSL relies on a chain of pre-signed certificates for server (and infrequently client) authentication. 23, *see* TCP, TLS & URL

Signposting A method to provide machines with explicit navigational link relations [Nottingham 2017] between Web resources, by providing HTTP `Link:` headers, HTML `<link>` elements or linkset JSON documens [Wilde 2020, Van de Sompel 2022]. 47, 74, 140, 187, 193, *see* HTTP, HTML, JSON & content-negotiation

Simple Knowledge Organization System (SKOS) SKOS is a lighweight RDF-based method of expressing vocabularies (terms and their definitions), frequently used for mapping between vocabularies and ontologies [Isaac 2009]. 37, 49, 192, 195

Specimen Data Refinery (SDR) Computational Workflow Platform to assist digitisation pipelines for specimens in natural history museum collections, combining aspects of FDO and RO-Crate. 95, 133, 134, 138, 143, 144, 148, 199

Transport Control Protocol (TCP) TCP is the most common connection protocol used on top of IP (TCP/IP), which opens a bidirectional channel where data is received by the application in the same order it was sent. The protocol includes handshakes and retried transmissions for reliable communication. 37, *see* IP

Transport Layer Security (TLS) Cryptographic protocol that largely replaces SSL for secure communication. Unlike SSL, TLS can be also be initiated from within an existing channel and gives the application further control of the encryption. 23, 37, *see* SSL

Uniform Resource Identifier (URI) Globally unique identifier string (e.g. `http://example.com/` or `urn:isbn:0451450523`). The *scheme* (`http`, `urn:isbn`) classifies the identifier. URIs are superset of URLs, as the scheme is not required to have an associated network protocol [Berners-Lee 2005]. Used as identifiers in RDF. 5, 23, 39, 47, 73, 80, 193, *see* URL & RDF

Uniform Resource Locator (URL) Globally unique string (e.g. `http://example.com/doc`) to identify a resource path (`/doc`) that can be retrieved (located) using the defined protocol (`http`) from the given domain name (`example.com`). Used for hyperlinks in HTML and most Web applications. 23, 37, 150, 188, *see* DNS, HTML & HTTP

Uniform Resource Name (URN) Globally unique identifier string. Subset of URIs which are *not* required to be locateable with a network protocol, e.g. `urn:isbn:9780062515865` or `urn:uuid:f7262a84-bd45-434e-a79f-32cbb55dc8b1`. URNs are to some extent deprecated in favour of HTTP-based PIDs, e.g. https://identifiers.org/isbn/9780062515865 23, 55, *see* URI & PID

Universally Unique Identifier (UUID) Globally unique identifier (also called *GUID*), a hexadecimal string representing a 128-bit number, e.g. `b91d75c1-891d-4305-85f9-7db73d9164da` [Leach 2005]. Typically generated from a random number generator. UUIDs can be generated independently and represent anything, but are not directly resolvable. 166, *see* URN

User Datagram Protocol (UDP) UDP is a connectionless overlay of IP, commonly used for scenarios where a short latency or high performance is more important than completeness of data packages (e.g. video streaming, gaming). 35, 37, *see* IP

Web Ontology Language (OWL) OWL is a method to express ontologies (classes, properties, hierarchies and inference rules). OWL is conceptually not RDF based, but considered part of the Semantic Web: OWL uses URIs as identifiers, is often used to define vocabularies for Linked Data, which are frequently published in RDFS-based serialisations [W3C 2012]. 2, 35, 49, 84, 188, 193, 195, 203, *see* RDF & LD

workflow In this thesis, a *computational workflow* is a machine-readable definition for executing a series of computational tools, typically executed using a workflow management system. 2, 10, 120

Workflow Management System (WfMS) Software or platform which can execute computational workflow definitions, and may manage data and provenance related to such executions. 2, 7, 121, 123, 137, 153, 157, 167, 178, 204, 207, *see* workflow

Workflow Run RO-Crate (WRROC) Extension of RO-Crate for representing computational workflow execution provenance 155, 157, 162, 172, 178, 196, 200, 202, 203, 207, *see* RO-Crate & WfMS

Yet Another Markup Language (YAML) YAML is a file format with the same data model as JSON, but with a more readable syntax, e.g. using indentation instead of quoted strings. https://yaml.org/ 165, 170, 194, 204, *see* JSON

ZIP A compressed archive file format. https://www.iana.org/assignments/media-types/application/zip 78, 80, 149, 150, 169

1
Introduction

Science is increasingly dependent on digital means, with computational methods used in almost all aspects of research, ranging from digitising plant specimens in herbariums [Thiers 2016], to molecular simulations of protein bindings for pharmacetical drug design [Śledź 2018].

Academics, government agencies and industry are now commonly making data publicly available under open licenses, feeding a broadening democratisation of science [Kitchin 2021] across social-economic borders[1], and expanding the potential for new multidiciplinary fields, commercialisation, citizen engagement and wider societal benefits [Bisol 2014].

Cloud-based computational infrastructures for "big data" are readily available for use with a wide range of open source software, enabling large scale secondary data analysis and detailed visualisations of research outputs [Hashem 2015].

However, in this accelerated ecosystem of Open Science, concerns have been raised about replicability of research findings [Ioannidis 2005], flagged as a "reproducibility crisis" [Baker 2016]. It is perhaps then ironic that the increased use of computers—with their inherently repeatable execution mechanisms—can negatively contribute to this crisis, as research publications do *not* commonly provide sufficient computational details such as code, data formats or software versions [Stodden 2016].

The increased focus on *reusability* of digital data and computational methods has been given the attention of funders and research communities. This led to the development of the FAIR principles for making data and their metadata *Findable, Accessible, Interoperable and Reusable*, i.e. retrievable and understandable for programmatic use [Wilkinson 2016].

One technological measure for achieving FAIR is using Linked Data, a set of practices for publishing and relating data on the Web using controlled vocabularies [Berners-Lee 2006], serialised using formats of the Resource Description Framework (RDF) [Schreiber 2014] and organised using the Web Ontology Language (OWL) [W3C 2012], however the combined complexity of these underlying *Semantic Web* technologies can hamper adoption by developers [Klímek 2019] and researchers who want to make their data available.

Computational workflows have been developed as ways to structure execution of software tools, for instance for scientific data analysis, so that, by using a Workflow Management System (WfMS), tool execution is reproducible, scalable and documented. For these purposes, workflow systems have become heavily adopted by some research fields such as life sciences, however the workflow definitions themselves are not yet commonly shared as part of scholarly outputs, and only gradually being recognised as a form of *FAIR Research Software* [Katz 2021b].

Research Object (RO) is a concept proposed for sharing composites of research artefacts, together with their history and related resources such as software, workflows and external references [Bechhofer 2013]. The initial implementations of RO heavily used ontologies, and required a tight integration with the workflow management systems, but has great potential for FAIR publication of any scholarly outputs.

[1] Although current open data practices do not benefit the Global South equally [Serwadda 2018].

The FAIR principles are widely referenced in Open Science literature, and nominally adapted by many research data repositories and funder policies—but how can they better be translated into practice by typical researchers and software developers which may be using workflow systems, but not know any Linked Data technologies?

This is the focus for this thesis, where I investigate *Linked Data approaches to implementing FAIR Research Objects and sharing reproducible Computational Workflows*.

1.1 Motivation – achieving FAIR research outputs

This section gives the motivation for the thesis, together with a brief background to inform the research questions in Section 1.2 on page 9. Further details on existing work are provided in Section 2 on page 16.

1.1.1 FAIR Principles

The FAIR Principles [Wilkinson 2016] were introduced to improve sharing and digital reuse of research outputs (”*data*”) as part of emerging open research practices. The main goals of FAIR are to support **F**indability, **A**ccessability, **I**nteroperability and **R**eusability, through machine-readable metadata and standardised publication methods for data, as quoted in Table 1.1.

In order to be **Findable**:	
F1	(Meta)data are assigned a globally unique and persistent *identifier*.
F2	Data are described with rich *metadata* (defined by R1 below).
F3	Metadata clearly and explicitly *include the identifier* of the data it describes.
F4	(Meta)data are *registered* or *indexed* in a searchable resource.
In order to be **Accessible**:	
A1	(Meta)data are *retrievable* by their identifier using a *standardized* communications protocol.
A1.1	The protocol is *open*, free, and universally implementable.
A1.2	The protocol allows for an *authentication* and *authorization* procedure, where necessary.
A2	*Metadata* are accessible, even when the *data are no longer available*.
In order to be **Interoperable**:	
I1	(Meta)data use a *formal*, accessible, shared, and broadly applicable *language for knowledge representation*.
I2	(Meta)data use *vocabularies* that follow FAIR principles.
I3	(Meta)data include qualified *references* to other (meta)data..
In order to be **Reusable**:	
R1	Meta(data) are richly described with a plurality of accurate and relevant *attributes*.
R1.1	(Meta)data are released with a clear and accessible *data usage license*.
R1.2	(Meta)data are associated with detailed *provenance*.
R1.3	(Meta)data meet domain-relevant community *standards*.

Table 1.1: FAIR Guiding Principles; adapted from [Wilkinson 2016], emphasis added in *italics*.

Although these guidelines are quite specific, they do not prescribe any particular technology or repository [Mons 2017]. Further formalizations of the FAIR principles include RDA’s FAIR Data Maturity Model [FAIR Maturity 2020, Bahui 2020]. FAIR has also been expanded beyond data, e.g. to cover software [Katz 2021b], computational workflows [Goble 2020], training materials [Garcia 2020a], machine learning models [Duarte 2023] and digital twins [Schultes 2022].

The FAIR principles have become highly influential for open research stakeholders [Jacobsen 2020], particularly in large research infrastructure initatives such as by the European Open Science Cloud (EOSC)[2] [Schouppe 2018], and increasing awareness and support for the principles by national Open Science policies and funders [Davidson 2019, Davidson 2022]. Implementation of the principles by platform developers and researchers have however raised many questions and practical challenges [Mons 2020, Riungu-Kalliosaari 2022].

For instance, in order to evaluate a given resource's *FAIRness*, additional technical constraints need to be assumed, such as use of particular formal vocabularies. *FAIR metrics* [Wilkinson 2018, Devaraju 2021] have recently become an area of active research, as different FAIR assessment tools may give a range of results for the same data resource, primarily based on which technical assumptions are made [Wilkinson 2022a, Verburg 2023].

Recently there have been increased emphasis on training and awareness on the FAIR principles [Shanahan 2021, Rocca-Serra 2023], and registries of standards and vocabularies [Sansone 2019]. However—with a general lack of skills in data management planning, inadequate (opaque) data formats, and not enough time investment to provide rich metadata—research data, even when shared through repositories, can become effectively "un-findable" or near impossible to reuse [Carballo-Garcia 2022].

From this current situation of we can identify several challenges with regards to finding practical ways for developers of Research Software to generate and consume FAIR data.

1.1.2 Existing approaches to implementing FAIR

The vision of the Semantic Web [Berners-Lee 1999] were proposed as a way to make structured data on the Web. This evolved into a Linked Data (LD) stack that uses logic-based ontologies, Web deployment of individually described resources, and cross-references between these resources with Uniform Resource Identifier (URI) identifiers. The *Semantic Web* can be considered as the ecosystem of such Linked Data resources, which can be queried, traversed and reasoned about.

Linked Data was seen early on as a possible mean to implementing the FAIR principles, and a large focus of initiatives like GO-FAIR[3] and Research Data Alliance[4] and the wider FAIR community has been to find ways to *FAIRify* existing data sources, such as developing domain-specific vocabularies and mappings, along with training and tooling to support these processes. FAIR publishing of datasets is encouraged using Data Catalog Vocabulary (DCAT) [Albertoni 2024], e.g. by the European Commission's Semantic Interoperability Community Europe (SEMIC)[5] and the larger Interoperable Europe[6] initiative.

[2] https://eosc.eu/

[3] https://www.go-fair.org/

[4] https://www.rd-alliance.org/

[5] https://joinup.ec.europa.eu/collection/semic-support-centre

[6] https://joinup.ec.europa.eu/interoperable-europe

There are now a large number of choices for Semantic Web technologies, serialisation formats, vocabularies, deployments and identifiers—motivating the proposal of *FAIR Implementation Profiles* [Schultes 2020] to document and guide technology decisions.

The field of Life Sciences was an early adopter of Linked Data, establishing training portals like FAIR Cookbook[7] [Rocca-Serra 2023], developing biomedical ontologies as indexed in BioPortal[8] [Whetzel 2011] (over 1300 as of 2024-05-18), and sharing practices at conferences like Semantic Web Applications for Health Care and Life Sciences (SWAT4HCLS)[9] active since 2008. The life science research infrastructure ELIXIR Europe[10] has over 170 training materials for FAIR[11] listed in its training portal TeSS (as of 2024-04-28), while the ELIXIR service FAIRsharing[12] [Sansone 2019] has over 1700 standards, 2100 databases and 250 policies (as of 2024-04-28) for FAIR sharing of research data[13].

A challenge for consumption of FAIR services in such a diverse landscape is thus how to support reliable *machine actionability*—making the data generally interpretable and typed sufficiently to allow invocation of pre-defined operations.

1.1.3 FAIR Digital Objects (FDO)

FAIR Digital Object (FDO) has been proposed as a machine-actionable ecosystem of scholarly outputs [Schultes 2019], and has now become a major initiative for realising the FAIR principles in a different way than the initial Semantic Web approach. FDO proponents envision a programmable mesh of strongly typed objects, which goes beyond the open data publication practices that the FAIR guidelines have popularised. For this, FDO aims to provide concrete constraints for systems, which lead to predictable machine actions.

The FDO guidelines[14] [Anders 2023a] and the more detailed FDO specifications [Anders 2023b] are largely conceptual in nature, with several demonstrated implementations [Wittenburg 2022a, Lannom 2022a] which in theory can operate side-by-side. Many of these, however, rely on novel or older network protocols [Reilly 2009, Sun 2003a] which are not particularly familiar to software developers, and not commonly supported by software libraries or frameworks.

This divergence from the more Web-centric "FAIR majority view", while sound from a technical perspective and promising with regards to predictable computational consumption, raises organisational challenges for wider adoption of FDOs, e.g. within EOSC and research infrastructures, and might be introducing a steeper learning curve than already exists for FAIR, particularly for developers of Research Software who are primarily interested in solving scientific challenges.

[7] https://faircookbook.elixir-europe.org

[8] https://bioportal.bioontology.org/

[9] https://www.swat4ls.org/

[10] https://elixir-europe.org/

[11] https://tess.elixir-europe.org/materials?q=FAIR

[12] https://fairsharing.org/

[13] It is worth noting that not all of these databases and standards are based on Linked Data methods, and may be supporting FAIR principles in a looser sense.

[14] Section 2.1.1 on page 17

Clearly the existing adoptions of Linked Data as-is would not present a coherent ecosystem for FDO machine-actionability, but it can be worth examining which aspects of the Web can benefit FDO development.

1.1.4 Research Software and Computational Workflows

A growing (if not majority) part of scientific analysis is now conducted using software and computational models. The concept of *Research Software Engineering* [Cohen 2020] has been established along with new professions *Research Software Engineer* [Baxter 2012] and *Data Scientist* [van der Aalst 2014]—researchers are not just using off-the-shelf software, but also combining multiple computational tools (e.g. in pipelines) and writing their own analytical source code (e.g. statistical R scripts) and simulations.

From this observation emerges the need to treat software as FAIR artefacts [Lamprecht 2019], following best practices for documentation [Lee 2018], open development [Prlić 2012] and ensuring Research Software (RS) is robust [Taschuk 2017] so it can be reused and cited as scholarly outputs [Smith 2016]. With this motivation, the principles of *FAIR Research Software* [Katz 2021b] have been established by the Research Data Alliance (RDA) working group FAIR for Research Software (FAIR4RS) [Barker 2022] and are gradually building traction, particularly in the life sciences. An example of a remaining challenge is how citations of Research Software can be practically propagated following their execution.

Sharing of Research Software according to these principles helps communicate the computational methods, expanding tremendously the potential for consumption, analysis and production of scientific data across organisations and their application to a broadening scope of research problems.

However, the *way* software is used for a particular analysis to reach a given scientific goal requires additional measures to make it *reproducible* [Stodden 2016, Sandve 2013]. *Computational Workflows* (or scientific workflows) can structure and automate data analysis pipelines so they are scalable, portable and explainable [Atkinson 2017], and as a side-effect of these features can significantly improve reproducibility [Cohen-Boulakia 2017].

Several challenges emerge when considering sharing of workflows as FAIR digital objects. For instance, a workflow composes multiple tools that themselves need to be shared. Data used by a workflow have their own attribution and licenses. The execution of a workflow produces many intermediate data, but understanding that data creation from the workflow definition alone requires deep knowledge about the particular Workflow Management System (WfMS).

1.1.5 Gathering scholarly outputs in Research Objects

The identified need for communicating computational methods through Research Software and workflows highlights that science must go beyond sharing of just data and metadata in order to achieve the FAIR principles. For a third-party researcher to fully take advantage of software and data, and to avoid delving further into the reproducibility crisis, the full set of contextual digital resources should be grouped and communicated as a scholarly unit.

Research Objects (ROs) [Bechhofer 2013] have been proposed as a mechanism to capture a range of diverse scholarly outputs in a single archivable item with detailed metadata. The RO concept was first realised using Semantic Web ontologies [myExperiment 2009, Belhajjame 2015]—these approaches primarily targetted long-term preservation of scientific workflows, utillised by RO as a mechanism to capture computational methods, augmented by the workflow inputs, outputs, workflow engine configuration and human-readable explanation of each step.

The principles of Research Objects extend far beyond workflows—however, early RO implementations mainly focused on capturing software [Goble 2018]. To some extent, the lack of wider adoption of ontology-based ROs can also be explained by Research Software Engineers (e.g. developers of molecular dynamics simulations) and platforms (e.g. repositories, data management systems) having a lack of familiarity with workflow systems or Semantic Web technology—or worse, they tried these technologies and then struggled [Carriero 2010, Tudorache 2020].

From this, a challenge is to make Linked Data technology approachable for developers who are best placed at implementing the FAIR principles, in platforms that are effectively making Research Objects.

1.2 Research Outline and Questions

Following the motivation in Section 1.1, this section elaborates my Research Questions (RQ) on three interlinked ideas:

1. Realization of the FAIR Digital Object concept using Web technologies.
2. Implementing FAIR Research Objects with an pragmatic use of Linked Data practices.
3. Unifying a FAIR Digital Object approach for computational workflows

1.2.1 Aims for FAIR Digital Objects on the Web (RQ1)

The Web is ubiquitous in modern software engineering [Taivalsaari 2021], used for everything from user interfaces, mobile applications and controlling devices, to enterprise cross-platform integrations, backend data processing and microservices, frequently utilising cloud computing which itself is controlled using Web technologies [Marinescu 2023].

The principles of FAIR Digital Objects (FDOs) seem important to achieve machine-actionable scholarly outputs, but several of these goals have an overlap with the motivations for the Semantic Web and Linked Data—yet it is not clear if changing from the Web stack to a different set of network protocols are necessary to achieve the FDO benefits.

A relevant research question therefore is:

RQ1: Can the promising FDO concept be realised using existing Web technology, taking into account the lessons learnt from the early Semantic Web developments and more recent Linked Data practices?

I address RQ1 in Chapters 2 and 3.

1.2.2 Aims for FAIR Research Objects (RQ2)

Following the lessons learnt on early Research Object (RO) implementations and the emerging FAIR principles, a new engagement between the RO and digital libraries communities started in 2018, where it was agreed to formulate a lightweight approach to Research Objects [Sefton 2018, Ó Carragáin 2019b] for the purpose of data packaging. From this initative, the updated aims of *FAIR Research Objects* can be summarised as:

- Describe and package data collections, datasets, software etc. with their metadata.
- Platform-independent object exchange between repositories and services.
- Support reproducibility and analysis: link data with codes and workflows.
- Transfer of sensitive/large distributed datasets with persistent identifiers.
- Aggregate citations and persistent identifiers.

- Propagate provenance and existing metadata.
- Publish and archive mixed objects and references.
- Reuse existing standards, but hide their complexity.

Following from these aims, the second research question is:

RQ2: Can a more pragmatic use of Linked Data practices better implement Research Objects for a wider developer audience, by using familiar Web technologies and give lightweight recommendations?

RQ2 is primarily addressed by Chapter 4.

1.2.3 Aims for FAIR Computational Workflows (RQ3)

There exists a plethora of workflow systems and languages [Leipzig 2021, Amstutz 2021], with recent efforts creating the Common Workflow Language [Crusoe 2022] as a standard representation with FAIR metadata capabilities[15] that is executable by multiple engines.

Notably, workflow definitions themselves can be considered FAIR scholarly outputs [Goble 2020]—*FAIR Computational Workflows* which are published in repositories like Dockstore [Yuen 2021] and WorkflowHub [Goble 2021]. One could consider computational workflows as a kind of FAIR Research Software [de Visser 2023], but by their nature workflows also *encourage the FAIR principles* (e.g. preparing a computational tool for a workflow system [Brack 2022a] may include publishing it in a container registry). Workflow systems are also useful for creating and consuming FAIR Digital Objects [Wittenburg 2022b], and in addition workflow systems commonly provide explicit provenance logs of their executions.

Approaches to describing workflow provenance in a machine-readable format were initially diverse [Cruz 2009], and later converged on the use of ontologies [Missier 2010], most notably using W3C PROV-O [Lebo 2013a] but with various specializations [Garijo 2011, Garijo 2012, Missier 2013, Belhajjame 2015, Cuevas-Vicenttín 2016].

The tendency for workflow provenance models to diverge may be down to differences in the execution semantics of different workflow systems—which if accurately reflected in provenance means further differences at this level. This in turn leads to incompatibility of provenance traces and lack of common tooling. In addition execution details may obscure the link from the computational processesses and the final workflow data outputs, which researchers ultimately care more about than the intricacies of the workflow engine.

The third research question from these considerations is therefore:

[15] https://www.commonwl.org/user_guide/topics/metadata-and-authorship.html

RQ3: Can a FAIR Digital Object approach for computational workflows unify machine-readable descriptions of Research Software, data and provenance, which can be consistently implemented by developers of different workflow management systems?

The multiple aspects of RQ3, as highlighted in this section, are adressed by Chapter 5.

1.3 Main Contributions

The contributions from this PhD include:

- An evaluation of FAIR Digital Objects and Linked Data, considering them from a developer perspective as distributed object systems.
- A Research Object implementation based on familiar Web technologies, adapted and extended by numerous research projects and software developers.
- A profile to capture provenance of computational workflow runs using this implementation, implemented by at least six workflow management systems.

These contributions have not evolved in isolation, but in co-development with multiple international collaborations (see Appendix A on page 210) across scientific diciplines.

1.4 Thesis Overview

Chapter 2 on page 15 gives the background of the concepts *FAIR Digital Object* (FDO) and *Linked Data*, including a brief history of the *Semantic Web*, followed by a critical analysis of these technologies and their use.

Chapter 3 on page 29 targets RQ1 and contributes a framework-based evaluation of Linked Data and FDO as possible architectures for implementing a distributed object system for the purpose of FAIR data publishing. The discussion in this chapter considers how the two approaches can benefit from each other's strengths.

Chapter 4 on page 75 addresses RQ2 by introducing the contribution of *RO-Crate*—a pragmatic data packaging mechanism using Linked Data standards to implement FDO and be extensible for domain-specific metadata.

Chapter 5 on page 119 considers RQ3 by exploring the relationship between Computational Workflows and FAIR practices using RO-Crate and FDO, with use cases from molecular dynamics and specimen digitization. The contribution of the *Workflow Run Crate profiles* is presented as an interoperable way to capture and publish workflow execution provenance.

Chapter 6 on page 185 summarises and discusses the contributions from this thesis, reflects on later third-party developments and concludes by evaluating the research questions.

1.5 Origins

Chapter 2 and Section 3.1 are based on journal article [Soiland-Reyes 2024b] (see Appendices A.4.1 and B.1.1). I am the main author of this manuscript.

> Stian Soiland-Reyes, Carole Goble, Paul Groth (2024):
> **Evaluating FAIR Digital Object and Linked Data as distributed object systems**.
> *PeerJ Computer Science* **10**:e1781
> https://doi.org/10.7717/peerj-cs.1781

Section 3.2 is based on [Soiland-Reyes 2022d] (see Appendices A.4.2 and B.1.2). I am the main author of this manuscript.

> Stian Soiland-Reyes, Leyla Jael Castro, Daniel Garijo, Marc Portier, Carole Goble, Paul Groth (2022):
> **Updating Linked Data practices for FAIR Digital Object principles**.
> *Research Ideas and Outcomes* **8**:e94501
> https://doi.org/10.3897/rio.8.e94501

Sections 4.1 and 4.3 are based on journal article [Soiland-Reyes 2022a] (see Appendices A.4.3, B.1.3 and B.1.5). I am the main author of this manuscript.

> 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 (2022):
> **Packaging research artefacts with RO-Crate**.
> *Data Science* **5**(2)
> https://doi.org/10.3233/DS-210053

Section 4.2 is based on [Soiland-Reyes 2022c] (see Appendices A.4.4 and B.1.4). I am the main author of this manuscript.

> Stian Soiland-Reyes, Peter Sefton, Leyla Jael Castro, Frederik Coppens, Daniel Garijo, Simone Leo, Marc Portier, Paul Groth (2022):
> **Creating lightweight FAIR digital objects with RO-Crate**.
> *Research Ideas and Outcomes* **8**:e93937
> https://doi.org/10.3897/rio.8.e93937

Section 5.1 is based on journal article [Soiland-Reyes 2022b] (see Appendices A.4.5 and B.1.6). I am the main author of this manuscript.

> Stian Soiland-Reyes, Genís Bayarri, Pau Andrio, Robin Long, Douglas Lowe, Ania Niewielska, Adam Hospital, Paul Groth (2022):
> **Making Canonical Workflow Building Blocks interoperable across workflow languages**.
> *Data Intelligence* **4**(2)
> https://doi.org/10.1162/dint_a_00135

Section 5.2 is based on journal article [Hardisty 2022] (see Appendices A.4.6 and B.1.7). I mainly contributed to Sections 5.2.2.2, 5.2.2.3, 5.2.4.1, 5.2.7 in this manuscript.

> Alex Hardisty, Paul Brack, Carole Goble, Laurence Livermore, Ben Scott, Quentin Groom, Stuart Owen, Stian Soiland-Reyes (2022):
> **The Specimen Data Refinery: A canonical workflow framework and FAIR Digital Object approach to speeding up digital mobilisation of natural history collections**.
> *Data Intelligence* **4**(2)
> https://doi.org/10.1162/dint_a_00134

Section 5.3 is based on [Woolland 2022] (see Appendices A.4.7 and B.1.8). I am the main author of this manuscript.

> Oliver Woolland, Paul Brack, Stian Soiland-Reyes, Ben Scott, Laurence Livermore (2022):
> **Incrementally building FAIR Digital Objects with Specimen Data Refinery workflows**.
> *Research Ideas and Outcomes* **8**:e94349
> https://doi.org/10.3897/rio.8.e94349

Section 5.4 is based on the preprint [Leo 2024] (see Appendices A.4.8 and B.1.9). I am the last author of this manuscript, and have mainly contributed to Sections 5.4.1, 5.4.5, 5.4.5.3, 5.4.5.4.

> Simone Leo, Michael R. Crusoe, Laura Rodríguez-Navas, Raül Sirvent, Alexander Kanitz, Paul De Geest, Rudolf Wittner, Luca Pireddu, Daniel Garijo, José M. Fernández, Iacopo Colonnelli, Matej Gallo, Tazro Ohta, Hirotaka Suetake, Salvador Capella-Gutierrez, Renske de Wit, Bruno de Paula Kinoshita, Stian Soiland-Reyes (2023):
> **Recording provenance of workflow runs with RO-Crate**.
> *arXiv* 2312.07852v1 [cs.DL]
> https://doi.org/10.48550/arXiv.2312.07852

This thesis also cites background material where I have contributed as co-author, provided as supplements[16] on the Web, see Appendix B.1.10 on page 233.

[16] https://s11.no/2023/phd/

2

FAIR Digital Objects and Linked Data

In this chapter, we discuss the related work with respect to FAIR Digital Objects and Linked Data. We do so by looking through the lens of development of these technologies over time, including future directions. This primarily motivates **RQ1** (on page 9) addressed by Chapter 3, but in addition both technologies are foundational for the implementations in Chapters 4 and 5.

2.1 FAIR Digital Object

The concept of **FAIR Digital Objects** [Schultes 2019] has been introduced as a way to expose research data as active objects that conform to the FAIR principles [Wilkinson 2016]. This builds on the *Digital Object* (DO) concept [Kahn 2006], first introduced by [Kahn 1995] as a system of *repositories* containing *digital objects* identified by *handles* [Sun 2003a] and described by *metadata* which may have references to other handles. DO was the inspiration for the [ITU-T X.1255] framework which introduced an abstract *Digital Entity Interface Protocol* for managing such objects programmatically, first realised by the Digital Object Interface Protocol (DOIP) [Reilly 2009].

In brief, the structure of a FAIR Digital Object (FDO) is to, given a Persistent Identifier (PID) such as a DOI, resolve to a *PID Record* that gives the object a *type* along with a mechanism to retrieve its *bit sequences*, *metadata* and references to further programmatic *operations* (Figure 2.1 on the facing page). The type of an FDO (itself an FDO) defines attributes to semantically describe and relate such FDOs to other concepts (typically other FDOs referenced by PIDs). The premise of systematically building an ecosystem of such digital objects is to give researchers a way to organise complex digital entities, associated with identifiers, metadata, and supporting automated processing [Wittenburg 2019].

This ecoystem is envisioned to consist of a wide variety of digital entities and contextual information ranging from software to articles to even descriptions of experimental infrastructures [Azeroual 2022]. Recently, it has been noted that the practical use of FDOs to achieve interoperability requires governance in particular with respect to assessing such interoperability [Wilkinson 2023a].

FDOs have been recognised by the European Open Science Cloud (EOSC[1]) as a suggested part of its Interoperability Framework [Kurowski 2021], in particular for deploying active and interoperable FAIR resources that are *machine actionable*[2]. Development of the FDO concept continued within Research Data Alliance (RDA[3]) groups and EU projects like GO-FAIR[4], concluding with a set of guidelines for implementing FDO [Bonino 2019]. The FAIR Digital Objects Forum[5] has since taken over the maturing of FDO through focused working groups which have currently

[1] https://eosc.eu/

[2] The concept of "machine actionable" is extended by FDO beyond the FAIR principles' premise of accessible data/metadata with retrievable vocabularies, in that programmatic invocation of operations on FAIR Digital Objects can be reliably coded in advance based on the information provided by the objects themselves (see Section 2.1.3 on page 20). The implications of considering FDOs as a distributed object system is explored further in Chapter 3.1 on page 31.

[3] https://www.rd-alliance.org/

[4] https://www.go-fair.org/

[5] https://fairdo.org/

drafted several more detailed specification documents [FDO Specs].

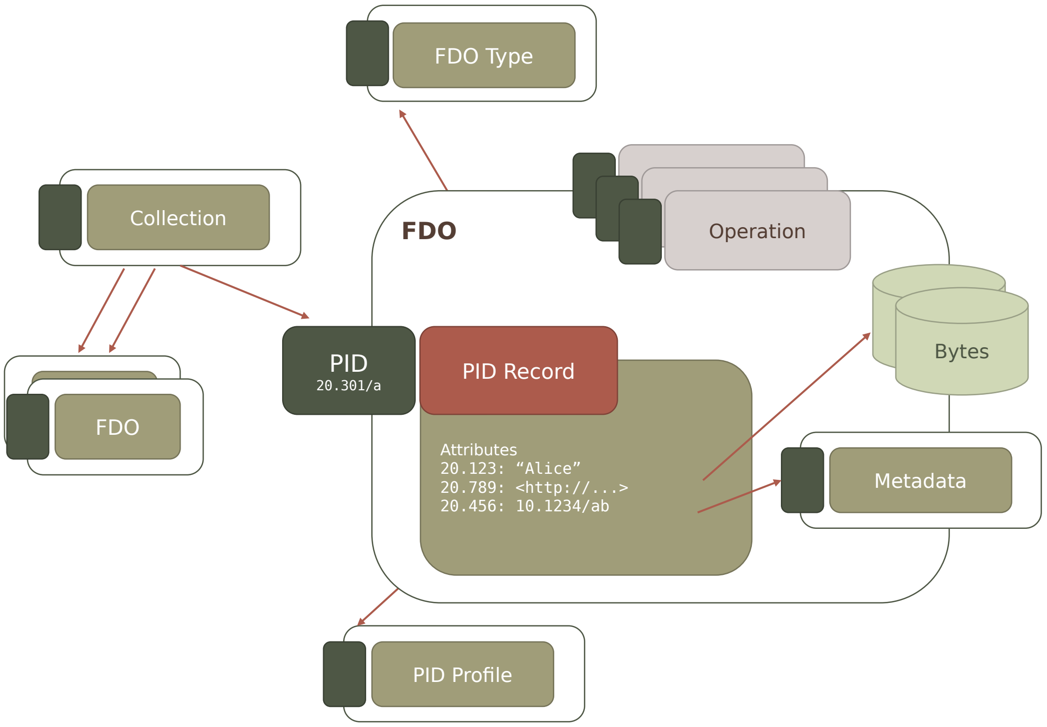


Figure 2.1: Idealised overview of a FAIR Digital Object. The persistent identifier (PID), (e.g. a Handle, DOI or permalink), refers to an FDO through a PID Record, which may reference downloadble bytes, and optionally additional metadata in another FDO. A series of operations are accessible from an FDO (for instance retrieving the bytes). Similar to in object-oriented programming, the FDO Type indicates which operations and attributes are applicable to an FDO. FDOs can be cross-related using the PIDs, a Collection is then another such FDO which aggregates other FDOs by reference. The configuration shown here is just one of many possible [Lannom 2022a], along with the choice of PID system, nature of the PID Record and metadata vocabularies, which are identified through an FDO Profile. In practice, some compromises from this idealised picture are taken depending on the implementation, for instance attribute keys may be simple strings rather than PIDs, and default operations are not explicitly declared.

2.1.1 FDO requirements

FDOs comply to a set of requirements [Anders 2023a], quoted below:

G1 FDOs should provide a path for long term infrastructure investments that is not tied to any particular technology stack.

G2 FDOs need to generate trust in accurate data survival over long periods of time, assuring researchers, funders, and developers that their significant effort in reusing them will be worthwhile.

G3 FDOs must offer compliance with the FAIR principles through measurable indicators of FAIRness.

G4 FDOs need to support machine actionability as being specified by FDO PR-MachineActionDef-2.0 [Weiland 2022b] or later.

G5 FDOs need to support the abstraction principle, i.e., abstracting away details that are not needed at the basic object management level. At that level there is no need to distinguish among different types such as data, metadata, software, semantic assertions, etc., for data management operations.

G6 FDOs need to support stable bindings among all information entities required for machine navigation of the global data space through the use of global, unique, and resolvable persistent identifiers.

G7 FDOs need to support encapsulation, such that operations can be associated with FDOs of all types.

G8 FDOs need to support technology independence, allowing implementations using different technologies.

G9 FDOs need to comply with minimal agreed standards, e.g., for movement of FDOs between systems, for interaction with FDOs, etc., to guarantee FDO interoperability across heterogeneous systems.

FDO-GR1 (*FDOF1*) A PID, standing for a globally unique, persistent and resolvable identifier, is assumed to be the basis for FAIR Digital Objects. Every FDO is assigned one or more PIDs.

FDO-GR2 (*FDOF2*) A PID resolves to a structured record (PID Record) compliant with a specified PID Profile which leads to resolution results that enable programmatic resolution from PID back to the FDO and its elements as specified by these requirements. PID Records represent the information characterising FDOs and together with their resolving PIDs they can themselves be FDOs.

FDO-GR3 (*FDOF3*) If an FDO contains a structured bit-sequence, the structured PID record includes at least a reference to the location(s) 1 where the bit-sequence encoding the content of a FAIR-DO (FDO) can be accessed as well as the type definition of the FDO. The structured record may also contain PIDs pointing to Metadata FDOs describing properties of the target FDO.

FDO-GR4 (*FDOF4*) The PID record needs to contain mandatory FDO attributes, may contain optional FDO attributes and attributes agreed upon by recognized communities. Values of attributes can be part of the PID record or they can be references. Expectations of which attribute values are contained within the PID record and which are references pointing from the record to external sources should be specified in the PID profile or definition of said attribute in a Data Type Registry.

FDO-GR5 (*FDOF5*) Each FDO identified by a PID can be accessed or operated on using an interface protocol by specifying the PID of a registered supported operation.

FDO-GR6 (*FDOF6*) Any basic FDO interface protocol offers standard Create, Read, Update, Delete (CRUD) operations on FDOs and a possibility to use extended/domain operations for specific FDO applications. The addition of an operation to list available extended/domain operation for specific FDO types is strongly recommended.

FDO-GR7 (*FDOF7*) The relations between FDO Types and supported operations are maintained in separate registries.

FDO-GR8 (*FDOF8*) Metadata can themselves be FDOs which describe the properties of the referenced FDO. They must be specified by a registered schema that refers to defined and registered metadata categories.

FDO-GR9 (*FDOF9*) Metadata can be of different types such as descriptive, domain specific, provenance, system, access permissions, transactions, etc.

FDO-GR10 (*FDOF10*) Metadata schemas are maintained by communities of practice and are FDOs. Such metadata schemas should therefore themselves follow FAIR principles.

FDO-GR11 (*FDOF11*) A collection of FDOs is also an FDO. The content of collection FDOs describes its construction using an agreed formal language which specifies the relationships of the constituent members. An FDO may be a member of several collections.

FDO-GR12 (*FDOF12*) Deletion of an FDO must lead to standardised and thus machine interpretable tombstone notes in metadata and PID records. The PID itself is not deleted.

FDO-GR13 The PID resolution and the FDO Layer information must be "machine actionable" i.e., are machine interpretable and belong to a type for which operations have been specified in symbolic grammar.

FDO-GR14 FDOs can be configured in different ways as long as the configurations are compliant with the FDO Specifications.

FDO-GR15 The granularity of FDOs is dependent on pragmatic utility decisions within the communities of practice Those communities define the level of useful entities to use in the corresponding application field.

Original *FDOF* identifiers from [Bonino 2019] are shown in italics above. The full list in [Anders 2023a] also include requirements for persistent identifiers (FDO-PIDR1 – FDO-PIDR6), attributes (FDO-FDOR1 – FDO-FDOR11) and resources (FDO-RESR1 – FDO-RESR2). The FDO specifications are detailed further in Section 2.1.3 on the next page.

2.1.2 FDO approaches

FDO is an evolving concept. A set of FDO Demonstrators [Wittenburg 2022a] highlights how current adopters are approaching implementations of FDO from different angles:

- Building on the Digital Object concept, using the simplified Digital Object Interface Protocol (DOIP) v2.0 [DONA 2018] specification, which detail how to exchange JavaScript Object Notation (JSON) objects through a text-based protocol[6] (usually TCP/IP over TLS). The main DOIP operations are retrieving, creating and updating digital objects. These are mostly realised using the reference implementation Cordra [Tupelo-Scheck 2022]. FDO types are registered in the local Cordra instance, where they are specified using JSON Schema [Wright 2022] and PIDs are assigned using the Handle system. Several type registries have been established.

- Following the Linked Data approach, but using the DOIP protocol, e.g. using JSON Linked Data (JSON-LD) and schema.org within DOIP in Materal Sciences archives [Riccardi 2022].

[6] For a brief introduction to DOIP 2.0, see [CNRI 2023b]

- Approaching the FDO principles from existing Linked Data practices on the Web, e.g. WorkflowHub use of RO-Crate and schema.org [Soiland-Reyes 2022c].

From this it becomes apparent that there is a potentially large overlap between the goals and approaches of FAIR Digital Objects and Linked Data, which we will cover in Section 2.2 on page 23.

2.1.3 An overview of upcoming FDO specifications

The FAIR Digital Object Forum [FDO] working groups have prepared detailed requirement documents [FDO Specs] setting out the path for realising FDOs, named *FDO Recommendations*. As of 2023-06-17, most of these documents are open for public review, while some are still in draft stages for internal review. As these documents clarify the future aims and focus of FAIR Digital Objects [Lannom 2022b], we provide a brief summary of each:

FAIR Digital Object Overview and Specifications [Anders 2023b] is a comprehensive overview of FAIR Digital Object specifications listed below. It serves as a primer that introduces FDO concepts and the remaining documents. It is accompanied by an FDO Glossary [Broeder 2022].

The **FDO Forum Document Standards** [Weiland 2022a] documents the recommendation process within the forum, starting at *Working Draft* (WD) status within the closed working group and later within the open forum, then *Proposed Recommendation* (PR) published for public review, finalised as *FDO Forum Recommendation* (REC) following any revisions. In addition, the forum may choose to *endorse* existing third-party notes and specifications.

The **FDO Requirement Specifications** [Anders 2023a] is an update of [Bonino 2019] as the foundational definition of FDO. This sets the criteria for classifying an digital entity as a FAIR Digital Object, allowing for multiple implementations. The requirements shown in Table 3.3 on page 39 are largely equivalent, but in this specification clarified and expanded with references to other FDO documents.

Machine Actionability [Weiland 2022b] sets out to define what is meant by *machine actionability* for FDOs. *Machine readable* is defined as elements of bit-sequences defined by structural specification, *machine interpretable* elements that can be identified and related with semantic artefacts, while *machine actionable* are elements with a type with operations in a symbolic grammar. The document largely describes requirements for resolving an FDO to metadata, and how types should be related to possible operations.

Configuration Types [Lannom 2022a] classifies different granularities for organising FDOs in terms of PIDs, PID Records, Metadata and bit sequences, e.g. as a single FDO or several daisy-chained FDOs. Different patterns used by current DOIP deployments are considered, as well as FAIR Signposting [Van de Sompel 2015, Van de Sompel 2022].

PID Profiles & Attributes [Anders 2022] specifies that PIDs must be formally associated with a *PID Profile*, a separate FDO that defines attributes required and recommended by FDOs following said profile. This forms the *kernel attributes*, building on recommendations from RDA's

PID Information Types working group [Weigel 2018]. This document makes a clear distinction between a minimal set of attributes needed for PID resolution and FDO navigation, which needs to be part of the *PID Record* [Islam 2023], compared with a richer set of more specific attributes as part of the *metadata* for an FDO, possibly represented as a separate FDO.

Kernel Attributes & Metadata [Weigel 2022] elaborates on categories of FDO Mandatory, FDO Optional and Community Attributes, recommending kernel attributes like `dateCreated`, `ScientificDomain`, `PersistencePolicy`, `digitalObjectMutability`, etc. This document expands on RDA Recommendation on PID Kernel Information [Weigel 2018]. It is worth noting that both documents are relatively abstract and do not establish PIDs or namespaces for the kernel attributes.

Granularity, Versioning, Mutability [Hellström 2022] considers how granularity decisions for forming FDOs must be agreed by different communities depending on their pragmatic usage requirements. The affect on versioning, mutability and changes to PIDs are considered, based on use cases and existing PID practices.

DOIP Endorsement Request [Schwardmann 2022a] is an endorsement of the DOIP v2.0 [DONA 2018] specification as a potential FDO implementation, as it has been applied by several institutions [Wittenburg 2022a]. The document proposes that DOIP shall be assessed for completeness against FDO—in this initial draft this is justified as *"we can state that DOIP is compliant with the FDO specification documents in process"* (the documents listed above).

Upload of FDO [Blanchi 2022] illustrates the operations for uploading an FDO to a repository, what checks it should do (for instance conformance with the PID Profile, if PIDs resolve). ResourceSync [ANSI/NISO Z39.99-2017] is suggested as one type of service to list FDOs. This document highlights potential practices by repositories and their clients, without adding any particular requirements.

Typing FAIR Digital Objects [Lannom 2022c] defines what *type* means for FDOs, primarily to enable machine actionability and to define an FDO's purpose. This document lays out requirements for how *FDO Types* should themselves be specified as FDOs, and how an *FDO Type Framework* allows organising and locating types. Operations applicable to an FDO is not predefined for a type; however, operations naturally will require certain FDO types to work. How to define such FDO operations is not specified.

Implementation of Attributes, Types, Profiles and Registries [Blanchi 2023] details how to establish FDO registries for types and FDO profiles, with their association with PID systems. This document suggest policies and governance structures, together with guidelines for implementations, but without mandating any explicit technology choices. Differences in use of attributes are examplified using FDO PIDs for scientific instruments, and the proto-FDO approach of DARIAH-DE[7] [Schwardmann 2022b].

It is worth pointing out that, except for the DOIP endorsement, all of these documents are

[7] https://de.dariah.eu/

conceptual, in the sense that they permit any technical implementation of FDO, if used according to the recommendations. Going forward a key strategy of the Forum is the use of profiles to help define specific attributes in metadata that are necessary for domains or application contexts. However, these are not yet fully implemented in the implementations considered here.

Existing FDO implementations [Wittenburg 2022a] are thus not fully consolidated in choices such as protocols, type systems and serialisations—this divergence and corresponding additional technical requirements mean that FDOs are not yet in a single ecosystem.

2.2 From the Semantic Web to Linked Data

In order to describe *Linked Data* as it is used today, we will start with an (opinionated) description of the evolution of its foundation, the *Semantic Web*.

2.2.1 A brief history of the Semantic Web

The **Semantic Web** was developed as a vision by Tim Berners-Lee [Berners-Lee 1999], at a time that the Web had already become widely established for information exchange, being a global set of hypermedia documents which are cross-related using universal links in the form of URLs. The foundations of the Web (e.g. URLs, HTTP, SSL/TLS, HTML, CSS, ECMAScript/JavaScript, media types) were standardised by W3C[8], Ecma[9], IETF[10] and later WHATWG[11]. The goal of Semantic Web was to further develop the machine-readable aspects of the Web, in particular adding *meaning* (or semantics) to not just the link relations, but also to the *resources* that the URLs identified, and for machines thus being able to meaningfully navigate across such resources, e.g. to answer a particular query.

Through W3C, the Semantic Web was realised with the Resource Description Framework (RDF) [Schreiber 2014] that used *triples* of subject-predicate-object statements, with its initial serialisation format [Lassila 1999] being RDF/XML (XML was at the time seen as a natural data-focused evolution from the document-centric SGML and HTML).

While triple-based knowledge representations were not new [Stanczyk 1987], the main innovation of RDF was the use of global identifiers in the form of URIs[12] as the primary identifier of the *subject* (what the statement is about), *predicate* (relation/attribute of the subject) and *object* (what is pointed to)—see Listing 2.1 on the following page. By using URIs not just for documents[13], the Semantic Web builds a self-described system of types and properties, where the meaning of a relation can be resolved by following its hyperlink to the definition within a *vocabulary*. By applying these principles as well to any kind of resource that could be described at a URL, this then forms a global distributed Semantic Web (Figure 2.2 on the next page).

[8] https://www.w3.org/standards/

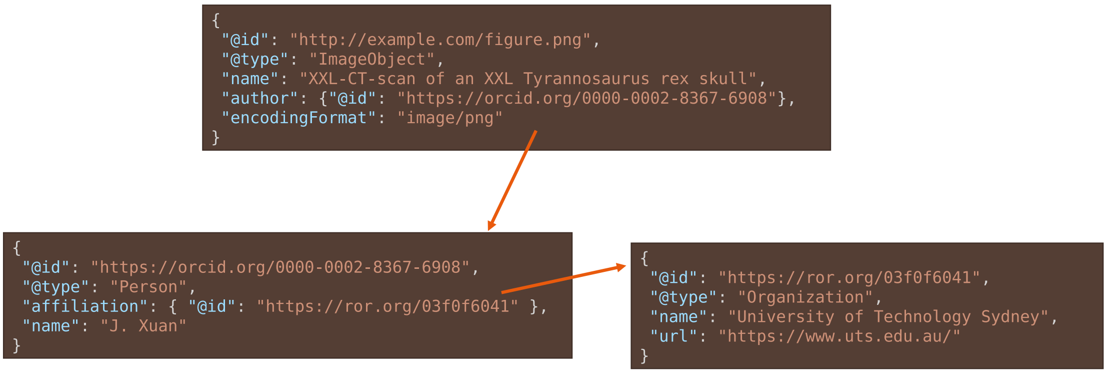
[9] https://www.ecma-international.org/

[10] https://www.ietf.org/standards/

[11] https://whatwg.org/

[12] URIs [Berners-Lee 2005] are generalised forms of URLs that include locator-less identifiers such as ISBN book numbers (URNs). The distinction between locator-full and locator-less identifiers has weakened in recent years [OCLC 2010], for instance DOI identifiers now are commonly expressed with the prefix `https://doi.org/` rather than as URN with `info:doi:` given that the URL/URN gap has been bridged by HTTP resolvers and the use of Persistent Identifiers (PIDs) [Juty 2011]. RDF 1.1 formats use Unicode to support IRIs [Dürst 2005], which extend URIs to include international characters and domain names.

[13] URIs can also identify *non-information resources* for any kind of physical object (e.g. people), such identifiers can resolve with `303 See Other` redirections to a corresponding *information resources* [Sauermann 2008].



```
{
 "@id": "http://example.com/figure.png",
 "@type": "ImageObject",
 "name": "XXL-CT-scan of an XXL Tyrannosaurus rex skull",
 "author": {"@id": "https://orcid.org/0000-0002-8367-6908"},
 "encodingFormat": "image/png"
}
```

```
{
 "@id": "https://orcid.org/0000-0002-8367-6908",
 "@type": "Person",
 "affiliation": { "@id": "https://ror.org/03f0f6041" },
 "name": "J. Xuan"
}
```

```
{
 "@id": "https://ror.org/03f0f6041",
 "@type": "Organization",
 "name": "University of Technology Sydney",
 "url": "https://www.uts.edu.au/"
}
```

Figure 2.2: Example of linked RDF resources. Each *resource* in an RDF graph has an identifier, here shown as absolute URIs, a type and a series of properties. A property value can either be a *literal* (e.g. ”Josiah Carberry”) or another resource (e.g. https://ror.org/03f0f6041). A graph is formed by such cross-references across resources. In the idealised Semantic Web, every URI would resolve to a description of its resource in RDF. In practice there can be misalignments of identifiers, vocabularies, resolution mechanisms, or simply lack of RDF adoption. Therefore, any RDF graph can describe any Web resource identified by its URI, and these descriptions, using an *open world assumption* [Drummond 2006], can be merged with other graphs describing the same resource. For brevity and comparison from later chapters this figure uses the newer RDF format JSON-LD [Sporny 2020], which can be expanded with context http://schema.org/ (not shown) to anchor types and properties as absolute URIs and generate corresponding RDF triples (Listing 2.1).

```
<http://example.com/figure.png> a <http://schema.org/ImageObject> .
<http://example.com/figure.png> <http://schema.org/name> "XXL-CT-scan of an XXL Tyrannosaurus
rex skull" .
<http://example.com/figure.png> <http://schema.org/author> <https://orcid.org/
0000-0002-1825-0097> .
<http://example.com/figure.png> <http://schema.org/encodingFormat> "image/png" .

<https://orcid.org/0000-0002-1825-0097> a <http://schema.org/Person> .
<https://orcid.org/0000-0002-1825-0097> <http://schema.org/name> "Josiah Carberry" .
<https://orcid.org/0000-0002-1825-0097> <http://schema.org/affiliation> <https://ror.org/
03f0f6041> .

<https://ror.org/03f0f6041> a <http://schema.org/Organization> .
<https://ror.org/03f0f6041> <http://schema.org/name> "University of Technology Sydney" .
<https://ror.org/03f0f6041> <http://schema.org/url> "https://www.uts.edu.au/" .
```

Listing 2.1: Example of RDF triples corresponding to Figure 2.2 after expansion with a JSON-LD context. In this example the properties and types are all using the same vocabulary [schema.org], in the traditional Semantic Web it is common to mix vocabularies. This listing uses the RDF syntax N-Triples [Carrothers 2014] where each line indicates *subject*, *predicate* and *object*. Notable here is the syntactical difference between an URI reference that is part of the graph `<https://ror.org/03f0f6041>` and a string literal `"https://www.uts.edu.au/"` which just happens to be a URI.

The early days of the Semantic Web saw fairly lightweight approaches with the establishment of vocabularies such as FOAF (to describe people and their affiliations) and Dublin Core (for bigbliographic data). Vocabularies themselves were formalised using RDFS or simply as human-readable HTML web pages defining each term. The main approach of this *Web of Data* was that a URI identified a *resource* (e.g. an author) with an HTML *representation* for human readers, along with a RDF representation for machine-readable data of the same resource. By using content negotiation[14] in HTTP, the same identifier could be used in both views, avoiding `index.html` vs `index.rdf` exposure in the URLs. The concept of *namespaces* gave a way to give a group of RDF resources with the same URI base from a Semantic Web-aware service a common *prefix*, avoiding repeated long URLs.

The mid-2000s saw large academic interest and growth of the Semantic Web, with the development of more formal representation system for ontologies, such as OWL [W3C 2012], allowing complex class hierarchies and logic inference rules following *open world* paradigm. (e.g. a *ex:Parent* is equivalent to a subclass of *foaf:Person* which must *ex:hasChild* at least one *foaf:Person*, then if we know *:Alice a ex:Parent* we can infer *:Alice ex:hasChild* [*a foaf:Person*] even if we don't know who that child is). More human-readable syntaxes for RDF such as Turtle evolved at this time, and conferences such as ISWC[15] [Horrocks 2022] gained traction, with a large interest in knowledge representation and logic systems based on Semantic Web technologies evolving at the same time.

Established Semantic Web services and standards include: SPARQL [W3C 2013] (pattern-based triple queries), named graphs[16] [Wood 2014] (triples expanded to *quads* to indicate statement source or represent conflicting views), triple/quad stores (graph databases such as OpenLink Virtuoso, GraphDB, 4Store), mature RDF libraries (including Redland RDF, Apache Jena, Eclipse RDF4J, RDFLib, RDF.rb, rdflib.js), and graph visualisation.

RDF is one way to implement *knowledge graphs*, a system of named edges and nodes[17] [Nurdiati 2008], which when used to represent a sufficiently detailed model of the world, can then be queried and processed to answer detailed research questions. The creation of RDF-based knowledge graphs grew particularly in fields like bioinformatics, e.g. for describing genomes and proteins [Goble 2008, Williams 2012]. In theory, the use of RDF by the life sciences would enable interoperability between the many data repositories and support combined views of the many aspects of bio-entities—however, in practice most institutions ended up making their own ontologies and identifiers, for what to the untrained eye would mean roughly the same. One can argue that the toll of adding the semantic logic system of rich ontologies meant that small, but fundamental, differences in opinion (e.g. *should a gene identifier signify just the particular DNA sequence letters, or those letters as they appear in a particular position on a human chromosome?*) led to large differences in representational granularity, and thus needed different identifiers.

[14] https://developer.mozilla.org/en-US/docs/Web/HTTP/Content_negotiation

[15] https://iswc2022.semanticweb.org/

[16] https://www.w3.org/TR/rdf11-concepts/#section-dataset

[17] In RDF, each triple represent an edge that is named using its property URI, and the nodes are subject/object as URIs, blank nodes or (for objects) typed literal values [Schreiber 2014].

Facing these challenges, thanks to the use of universal identifiers in the form of URIs, *mappings* could retrospectively be developed not just between resources, but also across vocabularies. Such mappings can themselves be expressed using lightweight and flexible RDF vocabularies such as SKOS [Isaac 2009] (e.g. `dct:title skos:closeMatch schema:name` to indicate near equivalence of two properties). Exemplifying the need for such cross-references, automated ontology mappings have identified large potential overlaps, like 372 definitions of `Person` [Hu 2011].

The move towards *Open Science* data sharing practices did from the late 2000s encourage knowledge providers to distribute collections of RDF descriptions as downloadable *datasets* [18], so that their clients can avoid thousands of HTTP requests for individual resources. This enabled local processing, mapping and data integration across datasets (e.g. Open PHACTS [Groth 2014]), rather than relying on the providers' RDF and SPARQL endpoints (which could become overloaded when handling many concurrent, complex queries).

With these trends, an emerging problem was that adopters of the Semantic Web primarily utillised it as a set of graph technologies, with little consideration to existing Web resources. This meant that links stayed mainly within a single information system, with little URI reuse even with large term overlaps [Kamdar 2017]. Just like *link rot* affect regular Web pages and their citations from scholarly communication [Klein 2014], a majority of described RDF resources in the Linked Open Data[19] (LOD) Cloud's gathering of more than thousand datasets do not actually link to (still) downloadable (*dereferenceable*) Linked Data [Polleres 2020]. Another challenge facing potential adopters is the plethora of choices, not just to navigate, understand and select to reuse the many possible vocabularies and ontologies [Carriero 2010], but also technological choices on RDF serialisation (at least 7 formats[20]), type system (RDFS [Guha 2014], OWL [W3C 2012], OBO [Tirmizi 2011], SKOS [Isaac 2009]), and deployment challenges [Sauermann 2008] (e.g. hash vs slash in namespaces, HTTP status codes and PID redirection strategies).

2.2.2 Linked Data: Rebuilding the Web of Data

The **Linked Data** (LD) concept [Bizer 2009] was kickstarted as a set of best practices [Berners-Lee 2006] to bring the Web aspect of the Semantic Web back into focus. Crucial to Linked Data is the *reuse of existing URIs*, rather than making new identifiers. This means a loosening of the semantic restrictions previously applied, and an emphasis on building navigable data resources, rather than elaborate graph representations.

Vocabularies like schema.org[21] evolved not long after, intended for lightweight semantic markup of existing Web pages, primarily to improve search engines' understanding of types and embedded data. In addition to several such embedded *microformats* [Open Graph, Sporny 2015,

[18] *Datasets* that distribute RDF graphs should not be confused with *RDF Datasets* used for partitioning *named graphs*, see https://www.w3.org/TR/rdf11-concepts/#section-dataset

[19] https://lod-cloud.net/

[20] https://www.w3.org/TR/rdf11-primer/#section-graph-syntax

[21] https://schema.org/

WHATWG 2023], we find JSON Linked Data (JSON-LD) [Sporny 2020] as a Web-focused RDF serialisation that aims for improved programmatic generation and consumption, including from Web applications. JSON-LD is as of 2023-05-18 used[22] by 45% of the top 10 million websites [W3Techs 2023].

Recently there has been a renewed emphasis to improve the *Developer Experience* [Verborgh 2018] for consumption of Linked Data, for instance RDF Shapes—expressed in SHACL [Kontokostas 2017] or ShEx [Baker 2019]—can be used to validate RDF Data [Labra Gayo 2017, Thornton 2019] before consuming it programmatically, or reshaping data to fit other models. While a varied set of tools for Linked Data consumptions have been identified, most of them still require developers to gain significant knowledge of the underlying Semantic Web technologies, which hampers adaption by non-LD experts [Klímek 2019], which then tend to prefer non-semantic two-dimensional formats such as CSV files.

A valid concern is that the Semantic Web research community has still not fully embraced the Web, and that the "final 20%" engineering effort is frequently overlooked in favour of chasing new trends such as Big Data and AI, rather than making powerful Linked Data technologies available to the wider groups of Web developers [Verborgh 2020]. One bridging gap here by the Linked Data movement has been "Linked Data by stealth" approaches such as structured data entry spreadsheets powered by ontologies [Wolstencroft 2011], the use of Linked Data as part of REST Web APIs [Page 2011], and as shown by the big uptake by publishers to annotate the Web using schema.org [Bernstein 2016], with vocabulary use patterns documented by copy-pastable JSON-LD examples, rather than by formalised ontologies or developer requirements to understand the full Semantic Web stack.

Linked Data provides technologies that have evolved over time to satisfy its primary purpose of data interoperability. The needs to embrace the Web and developer experience have been central lessons learned. In contrast, FDO is a new approach with many different potential paths forward, and having a partial overlap with the aims of Linked Data.

[22] Presumably this large uptake of JSON-LD is mainly for the purpose of Search Engine Optimisation (SEO), with typically small amounts of metadata which may not constitute Linked Data as introduced above; however, this deployment nevertheless constitutes machine-actionable structured data.

3

Comparing FDO and Linked Data as FAIR implementations

To investigate **RQ1** (on page 9) this chapter evaluates both Linked Data and FAIR Digital Object (FDO) as ways to realize the FAIR principles. Section 3.1 compares the two approaches as global distributed object systems, and discusses what lessons can be learnt across the communities, taking into consideration the history covered by Section 2.

Section 3.2 proposes how the FDO principles can be achieved using Linked Data standards, which is explored further in the following chapters.

3.1 Evaluating FAIR Digital Object and Linked Data as distributed object systems

FAIR Digital Object (FDO) is an emerging concept that is highlighted by European Open Science Cloud (EOSC) as a potential candidate for building an ecosystem of machine-actionable research outputs. In this work we systematically evaluate FDO and its implementations as a global distributed object system, by using five different conceptual frameworks that cover interoperability, middleware, FAIR principles, EOSC requirements and FDO guidelines themselves.

We compare the FDO approach with established Linked Data practices and the existing Web architecture, and provide a brief history of the Semantic Web[1] while discussing why these technologies may have been difficult to adopt for FDO purposes. We conclude with recommendations for both Linked Data and FDO communities to further their adaptation and alignment.

3.1.1 Introduction

The FAIR principles [Wilkinson 2016] encourage sharing of scientific data with machine-readable metadata and the use of interoperable formats, and are being adapted by a wide range of research infrastructures. They have been recognised by the research community and policy makers as a goal to strive for [EU 2016]. In particular, the European Open Science Cloud (EOSC)[2] has promoted adaptation of FAIR data sharing of data resources across electronic research infrastructures [Mons 2017]. The EOSC Interoperability Framework [Kurowski 2021] puts particular emphasis on how interoperability can be achieved technically, semantically, organisationally, and legally—laying out a vision of how data, publication, software and services can work together to form an ecosystem of digital objects that are extensively described. Such descriptions for interoperability connect a range of information—from protocols and presentations, to hardware designs and scientific workflows, including extensive metadata of the information itself.

Specifically, the EOSC Interoperability framework highlights the emerging FAIR Digital Object (FDO) concept [Schultes 2019] as a possible foundation for building a semantically interoperable ecosystem to fully realise the FAIR principles beyond individual repositories and infrastructures. The FDO approach has great potential, as it proposes strong requirements for identifiers, types, access and formalises interactive operations on objects.

In other discourse, Linked Data [Bizer 2009] has been seen as an established set of principles based on Semantic Web technologies that can achieve the vision of the FAIR principles [Bonino 2016, Hasnain 2018]. Yet regular researchers and developers of emerging platforms for computation and data management are reluctant to adapt such a "FAIR Linked Data" approach fully [Verborgh 2020], opting instead for custom in-house models and JSON-derived formats from RESTful Web services [Meroño-Peñuela 2021a, Neumann 2021]. While such focus on simplicity allows rapid development and highly specialised services, it raises wider concerns

[1] In this thesis moved to Section 2 on page 16 as background information on Linked Data and FDO.
[2] https://www.eosc.eu/

about interoperability [Turcoane 2014, Wilkinson 2022b].

One challenge that may, perhaps counter-intuitively, steer developers towards a not-invented-here mentality [Stefi 2015b, Stefi 2015a] when exposing their data on the Web is the heterogeneity and apparent complexity of Semantic Web approaches themselves [Meroño-Peñuela 2021b].

These approaches—FDO and Linked Data—thus, form two of the major avenues for allowing developers and the wider research community to achieve the goal of FAIR data. Given their importance, in this article we compare FAIR Digital Objects with Linked Data and the Web architecture in the context of the discourse around FAIR data.

Concretely, the contribution of this paper is a **systematic comparison between FDO and Linked Data using 5 different conceptual frameworks** that capture different perspectives on interoperability and readiness for implementation.

In Chapter 2 on page 16 we gave a background primer on FDO and Linked Data to provide a foundation for this work. The rest of this article is organised as follows: In the Method Section 3.1.2, we introduce the conceptual frameworks we use for comparison. Subsequently, in the Results Section 3.1.3 on the facing page, we systematically step through the outcomes of applying these frameworks to both FDO and Linked Data. For each framework, we derive key observations. We end in Section 3.1.4 on page 67 with a discussion of these results and their implications for both approaches and conclude.

3.1.2 Method

3.1.2.1 Comparing FDO and existing approaches

Our main motivation for this article is to investigate how FAIR Digital Objects may differ from the learnt experiences of Linked Data and the Web. We also aim to reflect back from FDO's motivation of machine-actionability to consider the Web as a distributed computational system.

To better understand the relationship between the FDO framework and other existing approaches, we use the following for analysis:

1. An Interoperability Framework and Distributed Platform for Fast Data Applications [Delgado 2016], which proposes quality measurements for comparing how frameworks support interoperability, particularly from a service architectural view.

2. The FAIR Digital Object guidelines [Bonino 2019], validated against its current implementations for completeness.

3. A Comparison Framework for Middleware Infrastructures [Zarras 2004], which suggest dimensions like openness, performance and transparency, mainly focused on remote computational methods.

4. Cross-checks against RDA's FAIR Data Maturity Model [Bahui 2020] to find how the FAIR principles are achieved in FDO, in particular considering access, sharing and openness.

5. EOSC Interoperability Framework [Kurowski 2021] which gives recommendations for technical, semantic, organisational and legal interoperability, particularly from a metadata perspective.

Conceptual frameworks 1, 3, 5 considers more general views of interoperability between systems, whereas frameworks 2 and 4 are developed specifically for addressing FAIR principles.

The reason for this wide-ranged comparison is to exercise the different dimensions that together form FAIR Digital Objects: Data, Metadata, Service, Access, Operations, Computation. We have left out further comparisons on type systems, persistent identifiers and social aspects as principles and practices within these dimensions are still taking form within the FDO community (as detailed in Section on page 20).

Some of these frameworks invite a comparison on a conceptual level, while others relate better to implementations and current practices. For these conceptual comparisons we consider FAIR Digital Objects and the Web broadly. For implementations we contrast the main FDO realisation using the DOIP v2 protocol [DONA 2018] against Linked Data as implemented in general practice[3].

For all our comparisons, our process was to perform a mapping between the relevant specifications and/or implementation and the given conceptual model through detailed reading of the defining documents. We aim in all cases for traceability between the given specification and our mapping such that readers can validate our analysis.

3.1.3 Results

3.1.3.1 Considering FDO/Web as interoperability framework for Fast Data

The Interoperability Framework for Fast Data Applications [Delgado 2016] categorises interoperability between applications along 6 strands, covering different architectural levels: from *symbiotic* (agreement to cooperate) and *pragmatic* (ability to choreograph processes), through *semantic* (common understanding) and *syntactic* (common message formats), to low-level *connective* (transport-level) and *environmental* (deployment practices).

We have chosen to investigate using this framework as it covers the higher levels of the OSI Model [Stallings 1990] better with regards to automated machine-to-machine interaction (and thus interoperability), which is a crucial aspect of the FAIR principles. In Table 3.1 we use the interoperability framework to compare the current FAIR Digital Object approach against the Web and its Linked Data practices.

[3] For further background on FDO implemented with Linked Data see [Bonino 2020] and Section 3.2 on page 71

Table 3.1: Considering FDO and Web according to the quality levels of the Interoperability Framework for Fast Data [Delgado 2016].

Quality	FDO with DOIP	Web with Linked Data
Symbiotic: *to what extent multiple applications can agree to interact, align, collaborate or cooperate*	The purpose of FDO is to enable federated machine actionable digital objects for scholarly purposes, in practice this also requires agreement of compatibility between FDO types. FDO encourages research communities to develop common type registries to be shared across instances. In current DOIP practice, each service have their own types, attributes and operations. The wider symbiosis is consistent use of PIDs.	The Web is loosely coupled and encourages collaboration and linking by URL. In practice, REST APIs [Fielding 2000] end up being mandated centrally by dominant (often commercial) providers [Fielding 2017], and the clients are required to use each API as-is with special code per service. Use of Linked Data enables common tooling and semantic mapping across differences.
Pragmatic: *using interaction contracts so processes can be choreographed in workflows*	FDO types and operations enable detailed choreography (Canonical Workflows [CWFR 2021]). Attributes[4] `0.TYPE/DOIPOperation` has lightweight definition of operation, `0.DOIP/Request` or `0.DOIP/Response` may give FDO Type or any other kind of "specifics" (incl. human-readable docs). Semantics/purpose of operations not formalised (similar operations can be grouped with `0.DOIP/OperationReference`).	"Follow your nose" crawler navigation, which may lead to frequent dead ends. Operational composition, typically within a single API provider, documented by OpenAPI 3 [Miller 2021], schema.org Actions [schema actions], WSDL/SOAP [Liu 2007], but frequently just as human-readable developer documentation with examples.

Quality	FDO w/ DOIP	Web w/ Linked Data
Semantic: *ensuring consistent understanding of messages, interoperability of rules, knowledge and ontologies*	FDO semantic enable navigation and typing. Every FDO has a type. Types maintained in FDO Type registries, which may add additional semantics, e.g. the ePIC PID-InfoType for Model[5]. No single type semantic, Type FDOs can link to existing vocabularies & ontologies. JSON-LD used within some FDO objects (e.g. DISSCO Digital Specimen, NIST Material Science schema) [Wittenburg 2022a]	Lightweight HTTP semantics for authenticity/navigation. Semantic Type not commonly expressed on PID/header level, may be declared within Linked Data metadata. Semantic of type implied by Linked Data formats (e.g. OWL2, RDFS), although choice of type system may not be explicit.
Syntactic: *serialising messages for digital exchange, structure representation*	DOIP serialise FDOs as JSON, metadata commonly use JSON, typed with JSON Schema. Multiple byte stream attachments of any media type.	Textual HTTP headers (including any signposting), single byte stream of any media type, e.g. Linked Data formats (JSON-LD, Turtle, RDF/XML) or embedded in document (HTML with RDFa, JSON-LD or Microdata). XML was previously the main syntax used by APIs, JSON is now dominant.
Connective: *transferring messages to another application, e.g. wrapping in other protocols*	DOIP [DONA 2018] is transport-independent, commonly TLS TCP/IP port 9000, DOIP over HTTP [CNRI 2023a]	HTTP/1.1 TCP/IP port 80 [Fielding 1999]; HTTP/1.1+TLS, TCP/IP 443 [Rescorla 2000]; HTTP/2, as HTTP/1* but binary [Belshe 2022]; HTTP/3, like HTTP/2+TLS but UDP [Bishop 2022]

Quality	FDO w/ DOIP	Web w/ Linked Data
Environmental: *how applications are deployed and affected by its environment, portability*	Main DOIP implementation is *Cordra*[6], which can be single-instance or distributed[7]. Cordra storage backends[8] include file system, S3, MongoDB (itself scalable). Unique DOIP protocol can be hard to add to existing Web application frameworks, although proxy services have been developed (e.g. B2SHARE adapter).	HTTP services widely deployed in a myriad of ways, ranging from single instance servers, horizontally & vertically scaled application servers, to multi-cloud Content-Delivery Networks (CDN). Current scalable cloud technologies for Web hosting may not support HTTP features previously seen as important for Semantic Web, e.g. content negotiation and semantic HTTP status codes.

Observations Based on the analysis shown in Table 3.1, we draw the following conclusions:

The Web has already showed us how one can compose workflows of hetereogeneous Web Services [Wolstencroft 2013]. However, this is mostly done via developer or human interaction [Lamprecht 2021]. Similiarly, FDO does not enable automatic composition because operation semantics are not well defined. There is a question as to whether the extensive documentation and broad developer usage that is available for Web APIs could potentially be utilised for FDO.

A difference between Web technologies and FDO is the stringency of the requirements for both syntax and semantics. Whereas the Web allows many different syntactic formats (e.g. from HTML to XML, PDFs), FDO realised with DOIP requires JSON. On the semantic front, FDO mandates that every object have a well-defined type and structured form. This is clearly not the case on the Web.

In terms of connectivity and the deployment of applications, the Web has a plethora of software, services, and protocols that are widely deployed. These have shown interoperability. The Web standards bodies (e.g. IETF and W3C) follow the OpenStand principles [OpenStand 2017] to embrace openness, transparency, and broad consensus. In contrast, FDO has a small number of implementations and corresponding protocols, although with a growing community, as evidenced at the first international FDO conference [Loo 2022]. This is not to say that it is not worth developing further Handle+DOIP implementations in the future, but we note that the current FDO functionality can easily be implemented using Web technologies, even as DOIP-over-HTTP [CNRI 2023a].

[4] DOIP's predefined attributes, types and operations have Handle-like identifiers with prefixes `0.TYPE` and `0.DOIP`, these are however not registered in the Handle system.

[5] https://hdl.handle.net/21.11104/c1a0ec5ad347427f25d6

[6] https://www.cordra.org/

[7] https://www.cordra.org/documentation/configuration/distributed-deployment.html

[8] https://www.cordra.org/documentation/configuration/storage-backends.html

It is also a question as to whether a highly constrained protocol revolving around persistent identifiers is in fact necessary. For example, DOIs are mostly resolved on the web using HTTP redirects with the common `https://doi.org/` prefix, hiding their Handle nature as an implementation detail [DOI 2019].

3.1.3.2 Mapping of Metamodel concepts

The Interoperability Framework for Fast Data also provides a brief *metamodel* which we use in Table 3.2 to map and examplify corresponding concepts in FDO's DOIP realization and the Web using HTTP semantics [Fielding 2022].

From this mapping[9] we can identify the conceptual similarities between DOIP and HTTP, often with common terminology. Notable are that neither DOIP or HTTP have strong support for transactions (explored further in Section 3.1.3.4 on page 45), as well that HTTP has poor direct support for processes, as the Web is primarily stateless by design [Fielding 2000].

Table 3.2: Mapping the Metamodel concepts from the Interoperability Framework for Fast Data [Delgado 2016] to equivalent concepts for FDO and Web.

Metamodel concept	FDO/DOIP concept	Web/HTTP concept
Resource	FDO/DO	Resource
Service	DOIP service	Server/endpoint
Transaction	(not supported)	Conditional requests, `409 Conflict`
Process	Extended operations	(primarily stateless), `100 Continue`, `202 Accepted`
Operation	DOIP Operation	Method, query parameters
Request	DOIP Request	Request
Response	DOIP Response	Response
Message	Segment, `requestId`	Message, Representation
Channel	DOIP Transport protocol (e.g. TCP/IP, TLS). JSWS signatures.	TCP/IP, TLS, UDP
Protocol	DOIP 2.0, ++	HTTP/1.1, HTTP/2, HTTP/3
Link	PID/	URL

[9] An equivalent SKOS mapping [Isaac 2009] is provided as part of the RO-Crate for this article [Soiland-Reyes 2023a].

3.1.3.3 Assessing FDO implementations

The FAIR Digital Object guidelines [Bonino 2019] sets out recommendations for FDO implementations. Note that the proposed update to FDO specification [Anders 2023a] clarifies these definitions with equivalent identifiers[10] and relates them to further FDO requirements such as FDO Data Type Registries.

In Table 3.3 on the next page we evaluate completeness of the guidelines in two current realizations: 1) DOIPv2 [DONA 2018] and 2) Linked Data Platform (LDP) [Sporny 2014], as proposed by [Bonino 2020]. We provide our analysis of each realisation with respect to the FDO Guideline and also provide suggestions for that realisation to meet the given guideline

[10] Newer [Anders 2023a] renames `FDOF*` to `FDO-GR*` but follows same ordering. For a brief listing of the requirements, see Section 2.1.1 on page 17.

Table 3.3: Checking FDO guidelines [Bonino 2019, Anders 2023a] against its current implementations as DOIP [DONA 2018] and Linked Data Platform (LDP) [Bonino 2020], with suggestions for required additions.

FDO Guideline	DOIP 2.0	FDO suggestions	Linked Data Platform	LDP suggestion
G1: *invest for many decades*	Handle system stable for 20 years, DOIP 2.0 since 2017.	Ensure FDO types will not be protocol-bound as DOIP might be updated/replaced	HTTP stable for 30 years, Semantic Web for 20 years. `http://` URIs mostly replaced by `https://`.	Keep flexibility of RDF serialisation formats which may change more frequently
G2: *trustworthiness*	DOI/Handle trusted by all major academic publishers and data repositories. DOIP relatively unknown, in effect only one implementation.	Further promote DOIP and justify its benefits. Build tutorials and OSI open source implementations. Standardise DOIP-over-HTTP alternative.	JSON-LD used by half of all websites [W3Techs 2023], however previous bad experiences with Semantic Web may deter adopters	Ensure simplicity for end developers, rather than semantic overengineering. Example-driven documentation.
G3: *follows FAIR principles*	See Table 3.5 on page 53	Ensure all FAIR principles are covered, build complete examples.	Touched briefly, see Table 3.5 on page 53	Add explicit expression to show each FAIR principle fulfilled.
G4: *machine actionability*	CRUD and extension operations dynamically listed (see Table 3.4 on page 46)	Specify which operations should work for a given type, to reduce need for dynamic lookup. Specify input/output expectations formally (e.g. JSON Schema).	HTTP CRUD operations, Open API (see Table 3.4 on page 46)	Document operations so client can make subsequent HTTP calls.

FDO	DOIP 2.0	FDO suggestions	Linked Data Platform	LDP suggestion
G5: *abstraction principle*	Handle PIDs as abstraction base. DOIP operations can use any transport protocol.	Document transport protocols as FDOs, recommend which transport to use.	URI as abstraction base. Does not specify PID requirements.	Give stronger deployment recommendations.
G6: *stable binding between entities*	Machine-navigation through PIDs and operations specified per type. Unclear when metadata field is a PID or plain text.	Make datatype of fields explicit to support navigation.	Machine-navigation through URIs via properties and types. Unclear when URI should be followed or is just identifier, but always distinct from plain text.	
G7: *encapsulation*	Operations discovered at runtime (`0.DOIP/Op.ListOperations`).	Allow method discovery by type FDOs in advance, see [Lannom 2022c].	HTTP methods discovered at runtime (`OPTIONS`), indempotent methods attempted directly. Unsupported methods reported using LDP constraints to human-readable text.	Declare supported methods in advance, e.g. OpenAPI [Miller 2021]
G8: *technology independence*	In theory independent, in reality depends on single implementations of Handle system and DOIP	Encourage open source DOIP testbeds and lighter reference implementations	Multiple HTTP implementations, multiple LDP implementations. No FDOF implementations.	Develop demonstrator of FDOF usage based on existing LDP server.
G9: *standard compliance*	Handle [Sun 2003a], DOIP [DONA 2018]. FDO requirements not standardised yet.	Formalise standard process of FDO requirements [Weiland 2022a]	HTTP, LDP. However FDOF is not yet standardised.	Formalise FDOF from FDOF-SEM working group.

FDO	DOIP 2.0	FDO suggestions	Linked Data Platform	LDP suggestion
FDOF1: *PID as basis*	Extensive use of Handle system.	Clarify how local testing handles can be used during development, how “temporary” FDOs should evolve [Anders 2022]. Register `0.DOIP/*` and `0.FDO/*` as actual PIDs.	HTTP URLs as basis for identifiers, but they are frequently not persistent.	Add strong guidance for PID services like w3id and persistence policies [McMurry 2017].
FDOF2: *PID record w/ type*	Unspecified how to resolve from Handle to DOIP Service (!), in practice `10320/loc`, `0.TYPE/DOIPService`, `URL`, `URL_REPLICA`	Document requirements for PID Record	w3id/purl PIDs redirect without giving any metadata. Datacite DOIs content-negotiate to give registered metadata.	Add FAIR Signposting [Van de Sompel 2022] at PID provider for minimal PID record
FDOF3: *PID resolvable to bytestream & metadata*	Byte stream resolvable (`0.DOIP/Retrieve`), `includeElementData` option can retrieve bytestream or full object structure. No method/attribute defined for separate metadata, only directly in PID Record. Unclear meaning of multiple items and bytestream chunks.	Clarify expectations for multiple items. Recommend chunks to not be used.	URIs resolvable by default. Multiple ways to resolve metadata, unclear preference.	Add FAIR Signposting and preference order.
FDOF4: *Additional attributes*	Freetext attribute keys. Attributes should be defined for FDO type.	Require that attribute keys should be PIDs (or have predefined mapping to PIDs). Explicitly allow attributes not already defined in type.	All attributes individually identified. Any Linked Data attributes can be used by URI or with mapped prefix.	Clarify type expectations of required/recommended/optional attributes.

FDO	DOIP 2.0	FDO suggestions	Linked Data Platform	LDP suggestion
FDOF5: *Interface: operation by PID*	Extended operations use PID, but "pid-like" DOIP operations/types are not registered as handles.	Register `0.DOIP/*` and `0.FDO/*` as PIDs. Clarify that operations can be mapped to protocol directly.	CRUD operations used directly in HTTP (e.g. `PUT`). Unclear how to provide PID for additional operations.	Specify how additional operations should be called over HTTP.
FDOF6: *CRUD operations + extensions*	`0.DOIP/Op.Create`, `Op.Retrieve`, `Op.Update`, `Op.Delete` but also `0.DOIP/Op.Search`.	Document	`PUT`, `GET`, `POST`, `DELETE`, `PATCH`, `HEAD` – extension operations (e.g. WebDAV `COPY`) not used, resource patterns [Ekuan 2023] are used instead.	Document how operation resources can be discovered from an LPD container. Document search API.
FDOF7: *FDOF Types related to operations*	Not yet formalised, by DOIP discoverable on a given FDO rather than type. PR-TypingFDOs leaves this open.	Add explicit relation between type and operations	`OPTIONS` per LDP Resource, but not by type. Common types (`ldp:Resource`, `ldp:Container`) indicate LDP support, but are not required.	Always make LDP types explicit in FDO profile.
FDOF8: *Metadata as FDO, semantic assertions*	DOIP includes all metadata in PID Record. Separate Metadata FDO need custom property.	Specify a `0.FDO/metadata` or similar to point to Metadata FDOs.	Assertions are always with semantics, using RDF vocabularies. Unspecified how to find additional metadata resources, `rdfs:seeAlso` is common.	Use FAIR Signposting `describedby` link relation to additional metadata PIDs

FDO	DOIP 2.0	FDO suggestions	Linked Data Platform	LDP suggestion
FDOF9: *Different metadata levels*	Defines open-ended "Response Attributes" without namespaces, but mandated as "None" for all CRUD operations. Metadata would need to be bundled within custom FDO types or attributes. Unclear how levels are separated within single FDO representation (may need FDOF8).	Declare which metadata are expected within response attribute or within FDO object. Require PIDs for custom attributes. Define how alternate metadata levels can be represented separately.	Undefined how to handle multiple metadata granularities or domains, alternative LDP containers can present different views on same stored objects.	Define how to navigate to alternate views and their semantic implications, e.g. `owl:sameAs`
FDOF10: *Metadata schemas by community*	Metadata schemas are in practice managed on single CORDA server as local types, using JSON Schema.	Require types to be FDOs with registered PIDs, implement shared types.	Plethora of existing RDF vocabularies/ontologies managed by larger communities, e.g. OBO Foundry[11] [Smith 2007]	Rather document better how individual ad-hoc schemas can be started for prototypes.
FDOF11: *FDO collections w/ semantic relations*	Collection type undefined by DOIP. Informal use of `HAS_PARTS` Handle attribute (e.g. [Semmler 2022]).		LDP Containers required by specification, also user-created (eg. `BasicContainer`).	Clarify relation to other collections like DCAT 3 [Albertoni 2024], Schema.org Dataset[12], OAI-ORE [Lagoze 2008]
FDOF12: *Deleted FDO preserve PID w/ tombstone*	Tombstone for deleted resource undefined by DOIP. `0.DOIP/Status.104` status code does not distinguish "Not Found" or "Gone"	Formalise tombstone requirements with new FDO type	`410 Gone` recommended, but 404 `Not Found` common. No requirement for tombstone serialisation	Formalise tombstone requirements and serialisation

[11] https://obofoundry.org/
[12] https://schema.org/Dataset

A key observation from this is that simply using DOIP does not achieve many of the FDO guidelines. Rather the guidelines set out how a protocol like DOIPs should be used to achieve FAIR Digital Object goals. The DOIP Endorsement [Schwardmann 2022a] sets out that to comply, DOIP must be used according to the set of FDO requirement documents (details in Section 2.1.3 on page 20), and notes *Achieving FDO compliance requires more than DOIP and full compliance is thus left to system designers*. Likewise, a Linked Data approach will need to follow the same requirements to comply as an FDO implementation.

Observations

- G1 and G2 call for stability and trustworthiness. While the foundations of both DOIP and Linked Data approaches are now well established—the FDO requirements and in particular, how they can be implemented, are still taking shape and subject to change.
- Machine actionability (G4, G6) is a core feature of both FDOs and Linked Data. Conceptually they differ in the way types and operations are discovered, with FDO seemingly more rigorous. In practice, however, we see that DOIP also relies on dynamic discovery of operations and that operation expectations for types (FDOF7) have not yet been defined.
- FDO proposes that types can have additional operations beyond CRUD (FDOF5, FDOF6), while Linked Data mainly achieves this with RESTful patterns using CRUD on additional resources, e.g. `order/152/items`. These are mainly stylistics but affect the architectural view—FDOs have more of an object-oriented approach.
- FDO puts strong emphasis on the use of PIDs (FDOF1, FDOF2, FDOF3, FDOF5), but in current practice DOIP use local types, local extended operations (FDOF5) and attributes (FDOF4) that are not bound to any global namespace.
- Linked Data have a strong emphasis on semantics (FDOF8), and metadata schemas developed by community agreements (FDOF10). FDO types need to be made reusable across servers.
- While FDO recommends nested metadata FDOs (FDOF8, FDOF9), in practice this is not found (or linked with custom keys), particularly due to lack of namespaces and the favouring of local types rather than type/property re-use. Linked Data frequently have multiple representations, but often not sufficiently linked (link relation `alternate` [Nottingham 2017]) or related (`prov:specializationOf` from [Lebo 2013a]).
- FDO collections are not yet defined for DOIP, while Linked Data seemingly have too many alternatives. LDP has specific native support for containers.
- Tombstones for deleted resources are not well supported, nor specified, for either approach, although the continued availability of metadata when data is removed is a requirement for FAIR principles (see RDA-A2-01M in Table 3.5 on page 56).
- DOIP supports multiple chunks of data for an object (FDOF3), while Linked Data can support content-negotiation. In either case it can be unclear to clients what is the meaning

or equivalence of any additional chunks.

3.1.3.4 Comparing FDO and Web as middleware infrastructures

In this section, we take the perspective that FDO principles are in effect proposing a global infrastructure of machine-actionable digital objects. As such we can consider implementations of FDO as **middleware infrastructures** for programmatic usage, and can evaluate them based on expectations for client and server developers.

We argue that the Web, with its now ubiquitous use of REST API [Fielding 2000], can be compared as a similar global middleware. Note that while early moves for developing Semantic Web Services [Fensel 2011] attempted to merge the Web Service and RDF aspects, we are here considering mainly the current programmatic Web and its mostly light-weight use of 3 out of possible *5 stars Linked Data* [Hausenblas 2012].

For this purpose, we here utillise the Comparison Framework for Middleware Infrastructures [Zarras 2004] that formalise multiple dimensions of openness, scalability, transparency, as well as characteristics known from Object-oriented programming such as modularity, encapsulation and inheritance.

Table 3.4: Comparing FAIR Digital Object (with the DOIP 2.0 protocol [DONA 2018]) and Web technologies (using Linked Data) as middleware infrastructures [Zarras 2004]

Quality	FDO with DOIP	Web with Linked Data
Openness: *framework enable extension of applications*	FDOs can be cross-linked using PIDs, pointing to multiple FDO endpoints. Custom DOIP operations can be exposed, although it is unclear if these can be outside the FDO server. PID minting requires Handle.net prefix subscription, or use of services like Datacite[13], B2Handle[14].	The Web is inherently open and made by cross-linked URLs. Participation requires DNS domain purchase (many free alternatives also exists). PID minting can be free using PURL/ARK services, or can use DOI/Handle with HTTP redirects.
Scalability: *application should be effective at many different scales*	No defined methods for caching or mirroring, although this could be handled by backend, depending on exposed FDO operations (e.g. Cordra can scale to multiple backend nodes)	Cache control headers reduce repeated transfer and assist explicit and transparent proxies for speed-up. HTTP `GET` can be scaled to world-population-wide with Content-Delivery Networks (CDNs), while write-access scalability is typically manage by backend.
Performance: *efficient and predictable execution*	DOIP has been shown moderately scalable to 100 millions of objects, create operation at 900 requests/second. DOIP protocol is reusable for many operations, multiple requests may be answered out of order (by `requestId`). Multiple connections possible. Setup is typically through TCP and TLS which adds latency.	HTTP traffic is about 10% of global Internet traffic, excluding video and social networks [Sandvine 2022]. HTTP 1 connections are serial and reusable, and concurrent connections is common. HTTP/2 adds asynchronous responses and multiplexed streams [Belshe 2022] but still has TCP+TLS startup costs. For reduced latency, HTTP/3 [Bishop 2022] use QUIC [Iyengar 2021] rather than TCP, already adapted heavily (30% of EMEA traffic) of which Instagram & Facebook video is the majority of traffic [Joras 2020].

Quality	FDO w/ DOIP	Web w/ Linked Data
Distribution transparency: *application perceived as a consistent whole rather than independent elements.*	Each FDO is accessed separately along with its components (typically from the same endpoint). FDOs should provide the mandatory kernel metadata fields. FDOs of the same declared type typically share additional attributes (although that schema may not be declared). DOIP does not enforce metadata typing constraints, this need to be established as FDO conventions.	Each URL accessed separately. Common HTTP headers provide basic metadata, although it is often not reliable. A multitude of schemas and serializations for metadata exists, conventions might be implied by a declared profile or certain media types. Metadata is not always machine findable, may need pre-agreed API URI Templates [Gregorio 2012], content-negotiation [MDN 2023] or FAIR Signposting [Van de Sompel 2022].
Access transparency: *local/remote elements accessed similarly*	FDOs should be accessed through PID indirection, this means difficult to make private test setup. Commonly a fixed DOIP server is used directly, which permits local non-PID identifiers.	Global HTTP protocol frequently used locally and behind firewalls, but at risk of non-global URIs (e.g. `http://localhost/object/1`) and SSL issues (e.g. self-signed certificates, local CAs)
Location transparency: *elements accessed without knowledge of physical location*	FDOs always accessed through PIDs. Multiple locations possible in Handle system, can expose geo-info.	PIDs and URL redirects. DNS aliases and IP routing can hide location. Geo-localised servers common for large cloud deployments.
Concurrency transparency: *concurrent processing without interference*	No explicit concurrency measures. FDO kernel metadata can include checksum and date.	HTTP operations are classified as being stateless/idempotent or not (e.g. `PUT` changes state, but can be repeated on failure), although these constraints are occassionally violated by Web applications. Cache control, `ETag` (e.g. checksum) and modification date in HTTP headers allows detection of concurrent changes on a single resource.
Failure transparency: *service provisioning resilient to failures*	DOIP status codes, e.g. `0.DOIP/Status.104`, additional codes can be added as custom attributes	HTTP status codes[15] e.g. `404 Not Found`, specific meaning of standard codes can be documented in Open API[16]. Custom codes uncommon.

Quality	FDO w/ DOIP	Web w/ Linked Data
Migration transparency: *allow relocating elements without interfering application*	Update of PID record URLs, indirection through `0.TYPE/DOIPServiceInfo` (not always used consistently). No redirection from DOIP service.	HTTP `30x` status codes provide temporary or permanent redirections, commonly used for PURLs but also by endpoints.
Persistence transparency: *conceal deactivation/reactivation of elements from their users*	FDO requires use of PIDs for object persistence, including a tombstone response for deleted objects. There is no guarantee that an FDO is immutable or will even stay the same type (note: Cordra extends DOIP with version tracking[17]).	URLs are not required to persist, although encouraged [Berners-Lee 1998]. Persistence requires convention to use PIDs/PURLs and HTTP `410 Gone`. An URL may change its content, change in type may sometimes force new URLs if exposing extensions like `.json`. Memento [Van de Sompel 2013] expose versioned snapshots. WebDAV `VERSION-CONTROL` method [Clemm 2002] (used by SVN).
Transaction transparency: *coordinate execution of atomic/isolated transactions*	No transaction capabilities declared by FDO or DOIP. Internal synchronisation possible in backend for Extended operations.	Limited transaction capabilities (e.g. `If-Unmodified-Since`) on same resource. WebDAV locking mechanisms[18] [Dusseault 2007] with `LOCK` and `UNLOCK` methods.
Modularity: *application as collection of connected/distributed elements*	FDOs are inheritedly modular using global PID spaces and their cross-references. In practice, FDOs of a given type are exposed through a single server shared within a particular community/institution.	The Web is inheritently modular in that distributed objects are cross-referenced within a global URI space. In practice, an API's set of resources will be exposed through a single HTTP service, but modularity enables fine-grained scalability in backend.

Quality	FDO w/ DOIP	Web w/ Linked Data
Encapsulation: *separate interface from implementation. Specify interface as contract, multiple implementations possible*	Indirection by PID gives separation. FDO principles are protocol independent, although it may be unclear which protocol to use for which FDO (although `0.DOIP/Transport` can be specified after already contacting DOIP). Cordra supports native DOIP[19] [CNRI 2023b], DOIP over HTTP[20] [CNRI 2023a] and Cordra REST API[21]	HTTP/1.1 semantics can seemlessly upgrade to HTTP/2 and HTTP/3. `http` vs `https` URIs exposes encryption detail[22]. Implementation details may leak into URIs (e.g. `search.aspx`), countered by deliberate design of URI patterns [Berners-Lee 1998] and PIDs via Persistent URLs (PURL).
Inheritance: *Deriving specialised interface from another type*	DOIP types nested with parents, implying shared FDO structures (unclear if operations are inherited). FDO establishes need for multiple Data Type Registries (e.g. managed by a community for a particular domain). Semantics of type system currently undefined for FDO and DOIP, syntactic types can also piggyback of FDO type's schema (e.g. Cordra `$ref`[23] use of JSON Schema references[24] [Wright 2022])	Syntactically media type with multiple suffixes [Sporny 2023] (mainly used with `+json`), declaration of subtypes as profiles (RFC6906) [Wilde 2013]. In metadata, semantic type systems (RDFS [Guha 2014], OWL2 [W3C 2012], SKOS [Isaac 2009]). OpenAPI 3 [Miller 2021] inheritance and Polymorphism[25]. XML `xsd:schemaLocation` or `xsd:type` [Thompson 2012], JSON `$schema` [Wright 2022], JSON-LD `@context` [Sporny 2020]. Large number of domain-specific and general ontologies define semantic types, but finding and selecting remains a challenge.
Signal interfaces: *asynchronous handling of messages*	DOIP 2.0 is synchronous, in FDO async operations undefined. Could be handled as custom jobs/futures FDOs	HTTP/2 multiplexed streams[26] [Belshe 2022], Web Sockets [Rice 2022], Linked Data Notifications [Capadisli 2017], AtomPub [Gregorio 2007], SWORD [Jones 2022], Micropub [Parecki 2017], more typically ad-hoc jobs/futures REST resources
Operation interfaces: *defining operations possible on an instance, interface of request/response messages*	CRUD predefined in DOIP, custom operations through `0.DOIP/Op.ListOperations` (can be FDOs of type `0.TYPE/DOIPOperation`, more typically local identifiers like `"getProvenance"`)	CRUD predefined in HTTP methods[27] [Fielding 2014b], (extended by registration)[28], URI Templates [Gregorio 2012], OpenAPI operations[29] [Miller 2021], HATEOAS[30] incl. Hydra [Lanthaler 2021], schema.org Actions [schema actions], JSON HAL [Kelly 2016] & Link headers (RFC8288) [Nottingham 2017]

Quality	FDO w/ DOIP	Web w/ Linked Data
Stream interfaces: *operations that can handle continuous information streams*	Undefined in FDO. DOIP can support multiple byte stream elements (need custom FDO type to determine stream semantics)	HTTP 1.1 [Fielding 2014a] chunked transfer[31], HLS (RFC8216) [Pantos 2017], MPEG-DASH [ISO 23009-1]

[13] https://datacite.org/

[14] https://eudat.eu/services/userdoc/b2handle

[15] https://datatracker.ietf.org/doc/html/rfc7231#section-6.5

[16] https://swagger.io/docs/specification/describing-responses/

[17] https://www.cordra.org/documentation/design/object-versioning.html

[18] https://datatracker.ietf.org/doc/html/rfc4918#section-6

[19] https://www.cordra.org/documentation/api/doip.html

[20] https://www.cordra.org/documentation/api/doip-api-for-http-clients.html

[21] https://www.cordra.org/documentation/api/rest-api.html

[22] The `http` protocol (port 80) can in theory also upgrade [Khare 2000] to TLS encryption, as commonly used by *Internet Printing Protocol* (https://www.rfc-editor.org/rfc/rfc8010.html#section-8.2) for `ipp` URIs, but on the Web, best practice is explicit `https` (port 443) URLs to ensure following links stay secure.

[23] https://www.cordra.org/documentation/design/schemas.html#schema-references

[24] https://json-schema.org/draft/2020-12/json-schema-core.html#references

[25] https://spec.openapis.org/oas/v3.1.0#composition-and-inheritance-polymorphism

[26] https://datatracker.ietf.org/doc/html/rfc7540#section-5

[27] https://datatracker.ietf.org/doc/html/rfc7231#section-4.3

[28] https://www.iana.org/assignments/http-methods/http-methods.xhtml

[29] https://spec.openapis.org/oas/v3.1.0.html#operation-object

[30] HATEOAS: Hypermedia as the Engine of Application State [Fielding 2000], an important element of the REST architectural style.

[31] https://datatracker.ietf.org/doc/html/rfc7230#section-4.1

Observations Based on the analysis in Table 3.4 on page 46, we make the following observations:

- With respect to the aspect of *Performance*, it is interesting to note that the first version of DOIP [Reilly 2009] supported multiplexed channels similar to HTTP/2 (allowing concurrent transfer of several digital objects). Multiplexing was removed for the much simplified DOIP 2.0 [DONA 2018]. Unlike DOIP 1.0, DOIP 2.0 will require a DO response to be sent back completely, as a series of segments (which again can be split the bytes of each binary *element* into sized *chunks*), before transmission of another DO response can start on the transport channel. It is unclear what is the purpose of splitting a binary into chunks on a channel which no longer can be multiplexed and the only property of a chunk is its size[32].
- HTTP has strong support for scalability and caching, but this mostly assumes read-operations from static resources. FDO has no view on immutability or validity of retrieved objects, but this should be taken into consideration to support large-scale usage.
- HTTP optimisations for performance (e.g. HTTP/2, multiplexing) are largely used for commercial media distribution (e.g. Netflix), and not commonly used by providers of FAIR data
- Cloud deployment of Web applications give many middleware benefits (Scalability, Distribution, Access transparancy, Location transparancy)—it is unclear how DOIP as a custom protocol would perform in a cloud setting as most of this infrastructure assumes HTTP as the protocol.
- Programmatically the Web is rather unstructured as middleware, as there are many implementation choices. Usually it is undeclared what to expect for a given URI/service, and programmers follow documented examples for a particular service rather than automated programmatic exploration across providers. This mean one can consider the Web as an ecosystem of smaller middlewares with commonalities.
- Many providers of FAIR Linked Data also provide programmatic REST API endpoints, e.g. UNIPROT[33], ChEMBL[34], but keeping the FAIR aspects such as retrieving metadata in such a scenario may require combining different services using multiple formats and identifier conventions.

[32] Although it is possible with `0.DOIP/Op.Retrieve` to request only particular individual elements of an DO (e.g. one file), unlike HTTP's `Range` request, it is not possible to select individual chunks of an element's bytestream.

[33] https://www.uniprot.org/help/programmatic_access

[34] https://chembl.gitbook.io/chembl-interface-documentation/web-services

3.1.3.5 Assessing FDO against FAIR

In addition to having "FAIR" in its name, the FAIR Digital Object guidelines [Anders 2023a] also include *G3: FDOs must offer compliance with the FAIR principles through measurable indicators of FAIRness*.

Here we evaluate to what extent the FDO guidelines and its implementation with DOIP and Linked Data Platform (LDP) [Bonino 2020] comply with the FAIR principles [Wilkinson 2016][35]. Here we have used the RDA's FAIR Data Maturity Model [FAIR Maturity 2020] as it has decomposed the FAIR principles to a structured list of FAIR indicators [Bahui 2020], importantly considering *Data* and *Metadata* separately. In our interpretation for Table 3.5 on the facing page we have for simplicity chosen to interpret "data" in FDOs as the associated bytestream of arbitrary formats, with remaining JSON or RDF structures always considered as metadata.

[35] For a brief list of the principles, see Table 1.1 on page 4.

Table 3.5: Assessing RDA's FAIR Data Maturity Model [FAIR Maturity 2020, Bahui 2020] (first 2 columns) against the FDO guidelines [Bonino 2019], FDO implemented with the protocol DOIPv2 [DONA 2018], Linked Data Platform (LDP) [Bonino 2020] and examples from Linked Data practices in general. (— indicates *Unspecified*, may be possible with additional conventions)

FAIR ID	Indicator	FDO guidelines	FDO/DOIP	FDO/LDP	Linked Data examples
RDA-F1-01M	Metadata is identified by a persistent identifier	FDOF4	Optional *Metadata FDO* w/separate PID	Content-negotiation to URL, not required to be PID	Metadata typically don't have own PID
RDA-F1-01D	Data is identified by a persistent identifier	FDOF1	PIDs required (FDOF1). Handle, DOI.	FDOF-IR (Identifier Record). PID can be any URI	"Cool" URIs [Berners-Lee 1998], PURL services incl. `purl.org`, `w3id.org`
RDA-F1-02M	Metadata is identified by a globally unique identifier	FDOF4 FDOF8	Optional *Metadata FDO*, unspecified how to indicate	Content-negotiation to URL	Not required, content-negotiation can redirect to URL or `Content-Location`. FAIR Signposting.
RDA-F1-02D	Data is identified by a globally unique identifier	FDOF1	All FDOs have PIDs (FDOF1), DOIP uses Handle system	FDOF-IR (Identifier Record)	Always accessed by URL
RDA-F2-01M	Rich metadata is provided to allow discovery	FDOF2 FDOF4 FDOF8 FDOF9	FDO has key-value metadata. Unclear how to link to additional metadata.	FDOF-IR links to multiple metadata records	RDF-based metadata by content negotiation or FAIR Signposting. Embedded in landing page (RDFa).

FAIR ID	Indicator	FDO	FDO/DOIP	FDO/LDP	LD examples
RDA-F3-01M	Metadata includes the identifier for the data	—	`id` and `type` are required metadata elements PIDs, also implicit as requests must use PID	PID only required in FDOF-IR record.	PID inclusion typical, but often inconsistent (e.g. `www.example.com` vs `example.com`) or missing (use of <> as *this* subject)
RDA-F4-01M	Metadata is offered in such a way that it can be harvested and indexed	FDOF10	No, registries not required (except Data Type Registries). Handle registry only searchable by PID.	—	Not specified, several registries/catalogues for vocabularies/types (e.g. [NCBO]). Indexing by search engines if exposing HTML w/schema.org.
RDA-A1-01M	Metadata contains information to enable the user to get access to the data	FDOF3 FDOF6	Directly by DOIP, but not included in FDO metadata. `handle.net` HTTP resolution may redirect to landing page	Any property can point to URIs, but unclear if it is data	Common with clickable “follow your nose” URLs
RDA-A1-02M	Metadata can be accessed manually (i.e. with human intervention)	—	(Cordra HTML landing page from `handle.net` URIs)	Optional content-negotiation, e.g. by Apache Marmotta, OpenLink Virtuoso	HTTP content-negotiation to HTML is common
RDA-A1-02D	Data can be accessed manually (i.e. with human intervention)	—	(Cordra HTML landing page from `handle.net` URIs)	Optional content-negotiation	Direct download, HTML landing pages common for DOIs
RDA-A1-03M	Metadata identifier resolves to a metadata record	FDOF8+FDOF2	—	—	`Content-Location` or HTTP redirection may indicate metadata URI

FAIR ID	Indicator	FDO	FDO/DOIP	FDO/LDP	LD examples
RDA-A1-03D	Data identifier resolves to a digital object	FDOF2	Required, but frequently not directly resolvable	Recommended, but any URI acceptable	Resolvable HTTP/HTTPS URIs are most common, now infrequent URNs are not directly resolvable
RDA-A1-04M	Metadata is accessed through standardised protocol	G9 FDOF3	Retrievable from PID (FDOF3). Informal DOIP standard maintained by DONA Foundation	LDP standard maintained by W3C, HTTP standards maintained by IETF, FDO components resolved by informal proposals (custom vocabulary, extra HTTP methods) or HTTP content negotiation)	Formal HTTP standards maintained by IETF, HTTP content negotiation, informal FAIR Signposting
RDA-A1-04D	Data is accessible through standardised protocol	G9	(see above)	HTTP [Fielding 2022]	HTTP/HTTPS, FTP (now less common), GridFTP [Allcock 2005] (for large data), ARK [Kunze 2022]
RDA-A1-05D	Data can be accessed automatically (i.e. by a computer program)	G4 FDOF3 FDOF6	Required, but few client libraries	HTTP `GET`, content-negotiation for `fdof/object`	Ubiquitous, hundreds of HTTP libraries

FAIR ID	Indicator	FDO	FDO/DOIP	FDO/LDP	LD examples
RDA-A1.1-01M	Metadata is accessible through a free access protocol	G1 G8 G9	Partially realised: Handle system is open[36] protocol [Sun 2003b]. One server implementation [Handle], free[37]. One DOIPv2 implementation (Cordra)[38]: free under BSD-like license (not recognised as Open Source).	LDP is open W3C recommendation [Sporny 2014]. Multiple LDP implementations[40].	DNS, HTTP, TLS, RDF standards are open, free and universal, large number of Open Source clients and servers[41].
RDA-A1.1-01D	Data is accessible through a free access protocol	G9	(see above)	URI, DNS, HTTP, TLS	URI, DNS, HTTP, TLS. Non-free DRM may be used (e.g. subscription video streaming)
RDA-A1.2-01D	Data is accessible through an access protocol that supports authentication and authorisation	(FDOF9)	TLS certificates, `authentication` field (details unspecified)	Implied	HTTP authentication, TLS certificates
RDA-A2-01M	Metadata is guaranteed to remain available after data is no longer available	FDOF12	—	Unspecified, however FDOF-IR links to separate metadata records	—
RDA-I1-01M	Metadata uses knowledge representation expressed in standardised format	FDOF8	Required, but not currently defined	—	Always implied by use of RDF syntaxes.
RDA-I1-01D	Data uses knowledge representation expressed in standardised format	—	—	—	Common (e.g. HDF5, JSON, XML), yet common scientific data formats frequently not standardised

FAIR ID	Indicator	FDO	FDO/DOIP	FDO/LDP	LD examples
RDA-I1-02M	Metadata uses machine-understandable knowledge representation	FDOF8	Required	Optional RDF metadata with any vocabulary	Always implied by use of RDF syntaxes.
RDA-I1-02D	Data uses machine-understandable knowledge representation	G4 G7 FDOF2	No requirements on binary data formats	Only indirectly, LDP Basic Container[42] reference only information resources	Common, specially for scientific data formats
RDA-I2-01M	Metadata uses FAIR-compliant vocabularies	G3 FDOF10	Informally required	Unspecified, implied by use of RDF?	FAIR practices for LD vocabularies increasingly common, sometimes inconsistent (e.g. PURLs that don't resolve) or incomplete (e.g. unknown license)
RDA-I2-01D	Data uses FAIR-compliant vocabularies	—	—	—	Uncommon, except for some XML and RDF-embedding formats, e.g. Extensible Metadata Platform (XMP) [ISO 16684]
RDA-I3-01M	Metadata includes references to other metadata	FDOF8	Implied (attributes to PIDs), currently unspecified if given attribute is value or reference	—	By definition (Linked Data reference existing URIs [W3C 2015]), `rdfs:seeAlso`, FAIR signposting [Van de Sompel 2022] `describedby`

FAIR ID	Indicator	FDO	FDO/DOIP	FDO/LDP	LD examples
RDA-I3-01D	Data includes references to other data	G6 FDOF3 FDOF11	—	—	URL hyperlinks common in several formats (HTML, PDF, JSON, XML).
RDA-I3-02M	Metadata includes references to other data	G6 FDOF3 FDOF8	Implied from custom FDO type's attribute	LDP Direct Container members can be any resources	URI objects are frequently data references, may be indirect via PID
RDA-I3-02D	Data includes qualified references to other data	FDOF3 FDOF11	Only indirectly through FDO metadata	Indirectly through LDP membership	Uncommon: Link relations, FAIR Signposting
RDA-I3-03M	Metadata includes qualified references to other metadata	(FDOF3)	Qualification by attribute keys defined per FDO Type	LDP Direct Container[43]	Qualifications by property, PROV bundles [Lebo 2013b], schema.org/Role[44]
RDA-I3-04M	Metadata include qualified references to other data	(FDOF3)	Qualification by attribute keys defined per FDO type	LDP Indirect Container[45]	Qualifications by property, n-ary indirection (schema.org Role [Holland 2014], `prov:specializationOf` [Lebo 2013a], OAI-ORE Proxy [Lagoze 2008])

FAIR ID	Indicator	FDO	FDO/DOIP	FDO/LDP	LD examples
RDA-R1-01M	Plurality of accurate and relevant attributes are provided to allow reuse	FDOF4	Required. Kernel metadata attributes desired [Weigel 2022] but not assigned PIDs yet.	Unspecified. Multiple metadata records can allow multiple semantic profiles.	Large number of general and domain-specific vocabularies can make it hard to find relevant attributes. Rough consensus on kernel metadata: schema.org [schema.org], Dublin Core Terms [DCMI 2020], DCAT [Albertoni 2020], FOAF [Brickley 2014]
RDA-R1.1-01M	Metadata includes information about the licence under which the data can be reused	—	`licenseConditions` URL/PID in kernel metadata [Weigel 2022]	—	Dublin Core Terms `dct:license` frequently recommended, frequently not required, e.g. by DCAT 2[46] [Albertoni 2020]
RDA-R1.1-02M	Metadata refers to a standard reuse licence	—	—	—	SPDX[47] and Creative Commons[48] URIs common, identifiers often inconsistent
RDA-R1.1-03M	Metadata refers to a machine-understandable reuse licence	—	—	—	SPDX documents[49] uncommon

FAIR ID	Indicator	FDO	FDO/DOIP	FDO/LDP	LD examples
RDA-R1.2-01M	Metadata includes provenance information according to community-specific standards	FDOF9 FDOF10	Unspecified (some Cordra types add `getProvenance` methods). PID Kernel attributes?	—	Unspecified W3C PROV-O, PAV
RDA-R1.2-02M	Metadata includes provenance information according to a cross-community language	FDOF9 FDOF8	—	—	W3C PROV-O [Lebo 2013a], PAV [Ciccarese 2013], Dublin Core Terms [DCMI 2020]
RDA-R1.3-01M	Metadata complies with a community standard	FDOF10 FROR8	(Emerging, e.g. DiSSCo Digital Specimen [Hardisty 2022])	—	Common, e.g. DCAT 2 [Albertoni 2020], BioSchemas [Gray 2017]
RDA-R1.3-01D	Data complies with a community standard	(FDOF3)	—	—	Common, HTTP use registered IANA media types[50], additional scientific file formats frequently not standardised or identified
RDA-R1.3-02M	Metadata is expressed in compliance with a machine-understandable community standard	FDOF4 FDOF10	Recommended	—	Common practice for ontologies, specially in bioinformatics, e.g. BioPortal [NCBO], Darwin Core [Wieczorek 2012]

FAIR ID	Indicator	FDO	FDO/DOIP	FDO/LDP	LD examples
RDA-R1.3-02D	Data is expressed in compliance with a machine-understandable community standard	(FDOF2)	No, FDO is typed but data can be any bytestream	—	Occassionally, (e.g. GFF3[51], FITS[52], ESRI[53])

[36] The `Handle.net` system was previously covered by software patent US6135646A(https://patents.google.com/patent/US6135646A/en) which expired in 2013 (https://circleid.com/posts/20161025_selling_dona_snake_oil_at_the_itu#11461)

[37] The Handle.net public license[39] is not OSI-approved [OSI 022] as an open source license—it includes usage restrictions and requires Service Agreements. It is not a DOIP requirement to host a local Handle instance, e.g. EOSC provides the *B2HANDLE* service for acquiring Handle prefixes (https://sp.eudat.eu/catalog/resources/fc6b2d30-09cd-4c25-b71a-7bc6de77910c).

[38] https://www.cordra.org/

[40] https://www.w3.org/wiki/LDP_Implementations

[41] https://en.wikipedia.org/wiki/Comparison_of_web_server_software

[42] https://www.w3.org/TR/ldp/#dfn-linked-data-platform-basic-container

[43] https://www.w3.org/TR/ldp/#dfn-linked-data-platform-direct-container

[44] https://schema.org/Role

[45] https://www.w3.org/TR/ldp/#dfn-linked-data-platform-indirect-container

[46] https://www.w3.org/TR/vocab-dcat-2/#Property:distribution_license

[47] https://spdx.org/licenses/

[48] https://creativecommons.org/

[49] https://spdx.dev/resources/use/#documents

[50] https://www.iana.org/assignments/media-types/media-types.xhtml

[51] https://github.com/The-Sequence-Ontology/Specifications/blob/master/gff3.md

[52] https://fits.gsfc.nasa.gov/fits_standard.html

[53] https://www.loc.gov/preservation/digital/formats/fdd/fdd000280.shtml

Observations

- Linked Data in general is strong on metadata indicators, but LDP approach is weak as it has little concrete metadata guidance.

- FDO/DOIP are stronger on identifier indicators, while Linked Data approach for identifiers relies on best practices.

- Indicators on standard protocols (RDA-A1-04M, RDA-A1-04D, RDA-A1.1-01M, RDA-A1.1-01D) favour LDP's mature standards (HTTP, URI)—the DOIPv2 specification [DONA 2018] has currently only a couple of implementations and is expressed informally. The underlying Handle system for PIDs is arguably mature and commonly used by researchers (this article alone references about 80 DOIs), however DOIs are more commonly accessed as HTTP redirects through resolvers like https://doi.org/ and http://hdl.handle.net/ rather than the Handle protocol.

- RDA-A1-02M and RDA-A1-02D highlights access by manual intervention, which is common for http/https URIs, but also using above PID resolvers for DOIP implementation Cordra[54] (e.g. https://hdl.handle.net/21.14100/90ec1c7b-6f5e-4e12-9137-0cedd16d1bce), yet neither LDP, FDO nor DOIP specifications recommends human-readable representations to be provided

- Neither DOIP nor LDP require license to be expressed (RDA-R1.1-01M, RDA-R1.1-02M, RDA-R1.1-03M), yet this is crucial for re-use and machine actionability of FAIR data and metadata to be legal

- Machine-understandable types, provenance and data/metadata standards (RDA-R1.1-03M RDA-R1.3-02M, RDA-R1.3-02M, RDA-R1.3-02D) are important for machine actionability, but are currently unspecified for FDOs. [Blanchi 2023] explores possible machine-readable FDO types, however the type systems themselves have not yet been formalised. Linked Data on the other side have too many semantic and syntactic type systems, making it difficult to write consistent clients.

- Indicators for FAIR data are weak for either approach, as too much reliance is put on metadata. For instance in Linked Data, given a URL of a CSV file, what is its persistant identifier or license information? Signposting [Van de Sompel 2015] can improve findability of metadata using HTTP Link relations, which enable an FDO-like overlay for any HTTP resource. In DOIP, responses for bytestreams can include the data identifier: if that is a PID (not enforced by DOIP), its metadata is accessible.

- Resolving FDOs via Handle PIDs to the corresponding DOIP server is currently undefined by FDO and DOIP specifications. `0.TYPE/DOIPServiceInfo` lookup is only possible once DOIP server is known.

[54] https://www.cordra.org/

3.1.3.6 EOSC Interoperability Framework

The European Open Science Cloud (EOSC) is a large EU initiative to promote Open Science by implementing a joint research infrastructure by federating existing and new services and focusing on interoperability, accessability, best practices as well as technical infrastructure [Ayris 2016]. The EOSC Interoperability Framework (EOSC-IF) [Kurowski 2021] details[55] the principles for creating a common way to achieve interoperability between all digital aspects of research activities in EOSC, including data, protocols and software. The recommendations are realised through 4 layers, Technical (e.g. protocols), Semantic (e.g. metadata models), Organisational (e.g. recommendations) and Legal (e.g. agreements), with a particular aim to address the FAIR interoperability principles and building on the concept of FAIR Digital Objects.

As covered in our introduction in Section 3.1.1 on page 31, EOSC proposes FAIR Digital Objects as a way to improve interoperability, for instance invoked by scientific workflows, carried by metadata frameworks and semantic artefacts. Therefore, we here find it important to summarize how FDO and Linked Data can help satisfy the EOSC requirements.

In Table 3.6 we review the EOSC Interoperability Framework (EOSC IF) recommendations, and evaluate to what extent they are addressed by the principles of FDO and Linked Data or their common implementations.

Table 3.6: Assessing EOSC Interoperability Framework [Kurowski 2021, section 3.6] against the FDO guidelines [Bonino 2019] and Linked Data practices.

Layer	Recommendation	FDO	Linked Data
Technical	Open Specification	FDO specifications are semi-open, process gradually more transparent	Open and transparent standard processes through W3C & IETF
Technical	Common security & privacy framework	Unspecified	TLS for encryption, multiple approaches for single-sign-on (e.g. ORCID, Life Science Login). Privacy largely unspecified.
Technical	Easy SLAs for service providers	Unspecified	None
Technical	Access data in different formats	None formalised, custom operations or relations	Content-negotiation, `rel=alternate` relations

[55] EOSC-IF has since been expanded on by an EOSC report [Åkerström 2024], which references the preprint of this Section [Soiland-Reyes 2024b].

Layer	Recommendation	FDO	Linked Data
Technical	Coarse-grained/fine-grained search tools	Freetext `0.DOIP/Op.Search` on local DOIP, no federation	Coarse-grained e.g. Google Dataset Search[56], fine-grained (e.g. federated SPARQL) require detailed vocabulary/metadata insight
Technical	Clear PID policy	Strong FDO requirements, tends towards Handle system.	Not required, different communities set policies
Semantic	Clear definitions for concepts/metadata/schemas	Required by FDO requirements, but not yet formalised	Ontologies, SKOS, OWL
Semantic	Semantic artefacts w/ open licenses	All artefacts are PIDs, license not yet required by kernel metadata	Open License is best practice for ontology publishing
Semantic	Documentation for each semantic artefact	No direct rendering from FDO, no requirement for human-readable description	Ontology rendering, content-negotiation
Semantic	Repositories of artefacts	Required, but not formalised	Bioontologies, otherwise not usually federated
Semantic	Repositories w/ clear governance	Recommended	Largely self-governed repositories, if well-established may have clear governance.
Semantic	Minimal metadata model for federated discovery	Kernel metadata [Weigel 2022] based on RDA recommendations [Weigel 2018].	DCAT, schema.org, Dublin Core
Semantic	Crosswalks from minimal metadata model	FDO Typing recommends referencing existing type definitions, but not as separate crosswalks	Multiple crosswalks for common metadata models, but frequently not in semantic format

Layer	Recommendation	FDO	Linked Data
Semantic	Extensibility options for diciplinary metadata	Communities encouraged to establish own types	Extensible by design, domain-specific metadata may be at different granularity
Semantic	Clear protocols/building blocks for federation/harvesting of artefact catalogues	Collection types not yet defined	SWORD, OAI-PMH
Organisational	Interoperability-focused rules of participation recommendations	Recommended	Implied only by some communities, tendency to specialise
Organisational	Usage recommendations of standardised data formats	None	None—but common for metadata (e.g. JSON-LD)
Organisational	Usage recommendations of vocabularies	Recommended by community	Common (see RDMKit[57])
Organisational	Usage recommendations of metadata	Recommended by community	RO-Crate, Gray 2017
Organisational	Management of permanent organization names/functions	Handle owner, but unclear contact. Contact info in DOIP service provider	ROR. DCAT contacts.
Legal	Standardised human and machine-readable licenses	None	SPDX[58], but not that frequently used
Legal	Permissive licenses for metadata (CC0, CC-BY-4.0)	Undefined	Both CC0, CC-BY-4.0 common, e.g. in DCAT
Legal	Different licenses for different parts	Each part as separate FDO can have separate license	DCAT, RO-Crate, Named graphs for splitting metadata
Legal	Mark expired/inexistent copyright	Undefined	Unclear, semantics assume copyright valid

Layer	Recommendation	FDO	Linked Data
Legal	Mark orphaned data	Tombstone for deleted data, but no owner of DOIP server means FDO disappears	Frequently data and endpoint has no known maintainer, archiving in common repositories becoming common
Legal	List recommended licenses	Undefined	Best practice recommendations
Legal	Track license evolution for dataset	Undefined	Versioning with PAV/PROV/DCAT
Legal	Policy/guidance for patent/trade secrets violation	Undefined	Undefined, legal owner may be specified. ODRL[59] can express policies
Legal	GDPR compliance for personal data	Undefined	Undefined
Legal	Restrict access/use if legally required	By transport protocol (undefined by FDO/DOIP)	Diverging approaches, typically landing pages w/ auth&auth or click-thru
Legal	Harmonised terms-of-use	Undefined	Undefined
Legal	Alignment between EOSC and national legislation	Not applicable	Not applicable

Observations Firstly, we observe that the EOSC IF recommendations are at a high level, mainly affecting governance and practices by communities. This *Organizational* level is also highlighted by the FDO recommendations, for instance the FDO Typing [Lannom 2022c] propose a governance structure to recognize community-endorsed services. While these community aspects are not mandated by Linked Data practices, best practices have become established for aspects like ontology development [Norris 2021]. EOSC IF's *Technical* layer is likewise at a architecturally high level, such as service-level agreements, but also highlight PID policies which is strongly required by FDO, while Linked Data communities choose PID practices separately. The recommendations for the *Semantic* layer, are largely already implemented by Linked Data practices,

[56] https://datasetsearch.research.google.com/
[57] https://rdmkit.elixir-europe.org/metadata_management
[58] https://spdx.org/licenses/
[59] https://www.w3.org/TR/2018/REC-odrl-vocab-20180215/

yet for FDO mostly consist of encouragements. For instance *clear definitions of semantic concepts* is required by FDO guidelines, but how to technically define them has not been formalised by FDO specifications.

The *Legal* layer of interoperability is perhaps the one most emphasised by EOSC, by enabling collaboration across organizational barriers to joinly build a research infrastructure, but this is an area that both FDO and Linked Data are relatively weak in directly supporting. The EOSC IF recommendations in this layer are largely related to governance practices and metadata, for instance licensing, privacy and usage policies; yet these are also essential for cross-institutional and cross-repository access of FAIR objects.

Likewise, search and indexing is important FAIR aspect for Findability, but is poorly supported globally by FDO and Linked Data. Efforts such as Open Research Knowledge Graph (ORKG) [Jaradeh 2019], DataCite's PID Graph [Fenner 2019] and Google Knowledge Graph [Singhal 2012] have improved programmatic findability to some degree, however not significantly for domain-specific semantic artefacts, currently scattered across multiple semantic catalogues [Corcho 2023]. There is a strong role for organizations like EOSC to provide such broader registries, moving beyond scholarly output metadata federations. The EOSC Marketplace[60] has for instance recently been expanded to include training material, software and data sources.

3.1.4 Discussion

We have evaluated the FAIR Digital Object concept using multiple frameworks, and contrasted FDO against existing experiences from Linked Data on the Web. In this section we discuss the implications of this evaluation, and propose how these two approaches can be better combined.

3.1.4.1 Framework evaluation

Having considered FDO and the Web architecture as interoperability frameworks (3.1.3.1 on page 33), we observe that neither are magic bullets, but each bring different aspects of interoperability. The Web comes with a large degree of flexibility and openness, however this means interoperability can suffer as services have different APIs and data models, although with common patterns. This is also true for Linked Data on the Web, with many overlapping ontologies and frequent inconsistencies in resolution mechanisms; although somewhat alleviated in recent years by schema.org becoming common metadata model for semantic markup inline in Web pages. The Web is based on a common HTTP protocol which has remained stable architecturally throughout its 32 years of largely backwards-compatible evolution. FDO on the other side sets down multiple rigid rules for identifiers, types, methods etc. that are adventurous for interoperability and predictability for FAIR consumption. Yet there is a large degree of freedom in how the FDO rules can be implemented by a given community, for instance there is no common

[60] https://marketplace.eosc-portal.eu/

metadata model or identifier resolution mechanism, and DOIP is just one possible transport method for FDOs, which itself does not enforce these rules.

When evaluating FDO implementations against the FDO guidelines (3.1.3.3 on page 38) we see that several technical pieces and community practices still need to be developed and further defined, for instance the FDO type system, how to declare FDO actions, how to resolve persistent identifiers, or how to know which pattern of FDO composition is used. Achieving fully interoperable FAIR Digital Objects would require further convergence on implementation practices, and it is not given that this needs to diverge from the established Web architecture. It is not clear from FDO guidelines if moving from HTTP/DNS to DOIP/Handle as a way to expose distributed digital objects will benefit FAIR practitioners, when both approaches require additional equably implementable restrictions and conventions, such as using persistent identifiers or pre-defining an object's type.

Considering this, by comparing FDO and Web as middleware (3.1.3.4 on page 45) we saw that programmatic access to digital objects, a core promise of FDO, is not particularly improved by the use of the protocol DOIP as compared to HTTP, e.g. lack of concurrency and transparency. Recent updates to HTTP have added many features needed for large-scale usage such as video streaming services (e.g. caching, multiplexing, cloud deployments), and having the option to transparently apply these also to FDOs seems like a strong incentive. Many programmatic features for distributed objects are, however, missing or needing custom extensions in both aspects, such as transactions, asynchronous operations and streaming.

By assessing FDO against the FAIR principles (3.1.3.5 on page 52) we found that both FDO implementations are underspecified in several aspects (licences, provenance, data references, data vocabularies, metadata persistence). While there are implementations of each of these in general Linked Data examples, there is no single set of implementation guides that fully realizes the FAIR principles. *FAIRification* efforts like the FAIR Cookbook [Rocca-Serra 2023] and FAIR Implementation Profiles [Schultes 2020] are bringing existing practices together, but there remains a potential role for FDO in giving a coherent set of implementation practices that can practically achieve FAIR. Significant effort, also within EOSC, is now moving towards FAIR metrics [Devaraju 2021], which in practice need to make additional assumptions on how FAIR principles are implemented, but these are not always formalised [Wilkinson 2022a] nor can they be taken to be universally correct [Verburg 2023]. Given that most of the existing FAIR guides and assessment tools are focused on Web and Linked Data, it would be reasonable for FDO to then provide a profile of such implementation choices that can achieve best of both worlds.

EOSC has been largely supportive of FDO, FAIR and related services. By contrasting the EOSC Interoperability Framework (3.1.3.6 on page 63) with FDO, we found that there are important dimensions that are not solved at a technical level, but through organization collaboration, legal requirements and building community practices. FDO recommendations highlight community aspects, but at the same time the largest FAIR communities in many science domains are already producing and consuming Linked Data. Just as the Linked Data community has a challenge in convincing more research fields to use Semantic Web technologies, FDO currently need to build many new communities in areas that have shown interest in that approach (e.g. material science). It may be advantageous for both these effort to be aligned and jointly promoted under the EOSC umbrella.

3.1.4.2 What does FDO mean for Linked Data?

The FAIR Digital Object approach raises many important points for Linked Data practictioners. At first glance, the explicit requirements of FDOs may seem to be easy to furfill by different parts of the Semantic Web Cake [Berners-Lee 2000, slide 10], as has previously been proposed [Soiland-Reyes 2022d][61] . However, this deeper investigation, based on multiple frameworks, highlights that the openness and variability of how Linked Data is deployed can make it difficult to achieve the FDO goals without significant effort.

While RDF and Linked Data have been suggested as prime candidates for making FAIR data, we argue that when different developers have too many degrees of freedom (such as serialization formats, vocabularies, identifiers, navigation), interoperability is hampered—this makes it hard for machines to reliably consume multiple FAIR resources across repositories and data providers. Indeed, this may be one reason why the initial FDO effort steered away from Linked Data approaches, but now seems in a danger of opening the many same degrees of freedom within FDO.

We therefore identify the need for a new explicit FDO profile of Linked Data that sets pragmatic constraints and stronger recommendations for consistent and developer-friendly deployment of digital objects. Such a combination of efforts could utillise both the benefits of mature Semantic Web technologies (e.g. federated knowledge graph queries and rich validation) and data management practices that follow FDO guidance in order to grow an ecosystem of machine-actionable objects. It is beyond the scope of this work to detail such a profile, but we suggest the following potential key aspects:

- Use HTTP(S) as protocol.
- Use URIs as identifiers, with persistent identifier promises.
- Provide consistent identifier resolution that does not require heuristics.
- Common core metadata model.
- References are always URIs, and should be persistent identifiers.
- Types, attributes and actions are self-defined by their identifier.
- Use Web approaches directly where possible, rather than wrap in a new model.

The FAIR and Linked Data communities likewise need to recognize the need for simpler, more pragmatic approaches that make it easier for FAIR practitioners to adapt the technologies with ”just enough” semantics.

[61] Section 3.2 on page 71

3.1.5 Conclusion

In this work, we have considered FAIR Digital Objects (FDO) as a potential distributed object system for FAIR data and compared it with established Web approaches focusing on Linked Data. We have described the background of the Semantic Web and FAIR Digital Objects, and evaluated both using multiple conceptual frameworks.

We find that both FDO and Linked Data approaches can significantly benefit from each-other and should be aligned further. Namely, Linked Data proponents need to make their technologies more approachable, agreeing on predictable and consistent implementations of FAIR principles.

The FDO recommendations show that FAIR thinking in this regard need to move beyond data publishing and into machine actionability across digital objects, and with broader community consensus. As flexibility for extensions is a necessary ingredient alongside rigidity for core concepts, the FDO community likewise need to settle on directly implementable specifications rather than just guidelines, and avoid making similar mistakes learnt by the early Semantic Web adopters.

By implementing the goals of FAIR Digital Objects with the mature technology stack developed for Linked Data, EOSC research infrastructures and researchers in general can create and use FAIR machine-actionable research outputs for decades to come.

3.2 Updating Linked Data practices for FAIR Digital Object principles

Realization of FAIR Digital Object (FDO) has a great potential if combined with the mature technology stack of Linked Data and knowledge graphs.

Here I will briefly discuss how FDO principles can be achieved using existing standards that have powered the Web for the last 30 years. Using this mature approach can accelerate uptake of FDO by scholars and existing research infrastructures.

I will also reflect on how the Linked Data (LD) community can adapt to better welcome approaches like FDO.

3.2.1 Background

The *FAIR principles* [Wilkinson 2016] are fundamental for data discovery, sharing, consumption and reuse; however their broad interpretation and many ways to implement can lead to inconsistencies and incompatibility [Jacobsen 2020].

The European Open Science Cloud (EOSC) has been instrumental in maturing and encouraging FAIR practices across a wide range of research areas. Linked Data in the form of RDF[62] (Resource Description Framework (RDF)) is the common way to implement machine-readability in FAIR; however, the principles do not prescribe RDF or any particular technology [Mons 2017].

3.2.1.1 FAIR Digital Object

FAIR Digital Object (FDO) **FAIR Digital Object** (FDO) [Schultes 2019] has been proposed to improve researcher's access to digital objects through formalising their metadata, types, identifiers and exposing their computational operations, making them actionable FAIR objects rather than passive data sources.

FDO is a set of principles [Bonino 2019], implementable in multiple ways. Current realisations mostly use *Digital Object Interface Protocol* (DOIPv2) [DONA 2018], with the main implementation Cordra[63]. We can consider DOIPv2 as a simplified combination of object-oriented (CORBA, SOAP) and document-based (HTTP, FTP) approaches.

More recently, the FDO Forum[64] has prepared detailed recommendations[65] [FDO Specs], currently open for comments, including a DOIP endorsement [Schwardmann 2022a] and updated FDO requirements [Anders 2023a]. These point out **Linked Data** as another possible technology stack, which is the focus of this work.

[62] https://www.w3.org/TR/rdf11-primer/

[63] https://www.cordra.org/documentation/api/doip.html

[64] https://fairdo.org/

[65] See Section 2.1.3 on page 20

3.2.1.2 Linked Data

Linked Data[66] (LD) standards, based on the Web architecture, are commonplace in sciences like bioinformatics, chemistry and medical informatics—in particular to publish Open Data as machine-readable resources. LD has become ubiquitous on the general Web, the schema.org[67] vocabulary is used by over 10 million sites for indexing by search engines—43% of all websites[68] use JSON-LD[69].

Although LD practices align to FAIR [Hasnain 2018], they do not fully encompass active aspects of FDOs. The HTTP protocol is used heavily for applications (e.g. mobile apps and cloud services), with REST APIs of customised JSON structures[70]. Approaches that merge the LD and REST worlds include Linked Data Platform[71] (LDP), Hydra[72] and Web Payments[73].

3.2.2 Meeting FDO principles using Linked Data standards

Considering the potential of FDOs when combined with the mature technology stack of LD, here we briefly discuss how FDO principles[74] can be achieved using existing standards. The general principles (G1–G9) apply well: Open standards with HTTP being stable for 30 years, JSON-LD is widely used, FAIR practitioners mainly use RDF, and a clear abstraction between the RDF model with stable bindings available in multiple serialisations.

However, when considering the specific principles (FDOF1–FDOF12) we find that additional constraints and best practices need to be established—arbitrary LD resources cannot be assumed to follow FDO principles. This is equivalent to how existing use of DOIP is not FDO-compliant without additional constraints.

Namely, Persistent Identifiers (PIDs) [McMurry 2017] (FDOF1) are common in LD world (e.g. using http://purl.org/ or https://w3id.org/); however, they don't always have a declared type (FDOF2), or the PID may not even appear in the metadata. URL-based PIDs are resolvable (FDOF3), typically over HTTP using redirections and content-negotiation. One great advantage of RDF is that all attributes are defined semantic artefacts with PIDs (FDOF4), and attributes can be reused across vocabularies.

While Create, Read, Update, Delete (CRUD) operations (FDOF6) are supported by native HTTP operations (GET/PUT/POST/DELETE) as in LDP[75], there is little consistency on how to define operation interfaces in LD (FDOF5). Existing REST approaches like OpenAPI[76] [Miller 2021]

[66] https://www.w3.org/standards/semanticweb/data
[67] https://schema.org/
[68] https://w3techs.com/technologies/details/da-jsonld
[69] https://json-ld.org/
[70] https://json-schema.org/
[71] https://www.w3.org/TR/ldp/
[72] https://www.hydra-cg.com/
[73] https://www.w3.org/TR/webpayments-http-messages/
[74] FDO guidelines are listed in Section 2.1.1 on page 17.
[75] https://www.w3.org/TR/ldp/
[76] https://swagger.io/specification/

and URI templates [Gregorio 2012] are mature and good candidates, and should be related to defined types to support machine-actionable composition (FDOF7). HTTP error code *410 Gone* is used in tombstone pages for removed resources (FDOF12), although more frequent is *404 Not Found*.

Metadata is resolved to HTTP documents with their own URIss, but these frequently don't have their own PID (FDOF8). RDF-Star[77] and nanopublications [Kuhn 2021] give ways to identify and trace provenance of individual assertions.

Different metadata levels (FDOF9) are frequently developed for LD vocabularies across different communities (FDOF10), such as FHIR[78] for health data, Bioschemas[79] for bioinformatics and >1000 more specific bioontologies[80]. Increased declaration and navigation of *profiles* is therefore essential for machine-actionability and consistent consumption across FAIR endpoints.

Several standards exist for rich collections (FDOF11), e.g. OAI-ORE[81], DCAT[82], RO-Crate[83], LDP[84]. These are used and extended heterogeneously across the Web, but consistent machine-actionable FDOs will need specific choices of core standards and vocabularies. Another challenge is when multiple PIDs refer to "almost the same" concept in different collections—significant effort have created manual and automated semantic mappings [Baker 2013, de Mello 2022].

Currently the FDO Forum has suggested the use of LDP as a possible alternative for implementing FAIR Digital Objects [Bonino 2020], which proposes a novel approach of content-negotiation with custom media types.

3.2.3 Discussion

The Linked Data stack provides a set of specifications, tools and guidelines in order to help the FDO principles become a reality. This mature approach can accelerate uptake of FDO by scholars and existing research infrastructures such as the European Open Science Cloud (EOSC).

However, the amount of standards and existing metadata vocabularies poses a potential threat for adoption and interoperability. Yet, the challenges for agreeing on usage profiles apply equally to DOIP as LD approaches.

We have worked with different scientific communities to define RO-Crate [Soiland-Reyes 2022a], a lightweight method to package research outputs along with their metadata. While RO-Crate's use of schema.org shows just one possible metadata model, it's powerful enough to be able to express FDOs, and familiar to web developers.

[77] https://w3c.github.io/rdf-star/
[78] http://hl7.org/fhir/
[79] https://bioschemas.org/
[80] https://bioportal.bioontology.org/ontologies
[81] https://www.openarchives.org/ore/
[82] https://www.w3.org/TR/vocab-dcat-3/
[83] https://www.researchobject.org/ro-crate/
[84] https://www.w3.org/TR/ldp/

We have also used FAIR Signposting [Van de Sompel 2022] with HTTP `Link:` headers as a way to support navigation to the individual core properties of an FDO (PID, type, metadata, licence, bytestream) that does not require heuristics of content-negotiation and is agnostic to particular metadata vocabularies and serialisations.

We believe that by adopting Linked Data principles, we can accelerate FDO today—and even start building practical ways to assist scientists in efficiently answering topical questions based on knowledge graphs.

4
RO-Crate

This chapter introduces *RO-Crate*, a pragmatic method of packaging data alongside structured metadata that is inline with the FAIR principles. This has been implemented to investigate **RQ2** (on page 10).

Section 4.1 on the facing page describes the RO-Crate purpose, community effort and tooling and demonstrates how RO-Crate has been applied.

Section 4.2 on page 109 shows how RO-Crate can be used to achieve the FDO principles covered in Chapter 3.

Section 4.3 on page 113 contributes a formal definition of RO-Crate using first order logic.

Supplementary material that may assist readers of this chapter includes the motivation of RO-Crate, a lightweight approach to Research Object data packaging[1] [Ó Carragáin 2019b].

RO-Crate builds on the long history of *Research Objects*, which is covered by earlier works [Bechhofer 2013, Belhajjame 2015, Goble 2018] and the Wf4Ever project[2].

[1] https://s11.no/2019/phd/ro-crate/

[2] https://s11.no/2020/archive/wf4ever/

4.1 Packaging research artefacts with RO-Crate

An increasing number of researchers support reproducibility by including pointers to and descriptions of datasets, software and methods in their publications. However, scientific articles may be ambiguous, incomplete and difficult to process by automated systems. In this paper we introduce RO-Crate, an open, community-driven, and lightweight approach to packaging research artefacts along with their metadata in a machine readable manner. RO-Crate is based on schema.org annotations in JSON-LD, aiming to establish best practices to formally describe metadata in an accessible and practical way for their use in a wide variety of situations.

An Research Object Crate (RO-Crate) is a structured archive of all the items that contributed to a research outcome, including their identifiers, provenance, relations and annotations. As a general purpose packaging approach for data and their metadata, RO-Crate is used across multiple areas, including bioinformatics, digital humanities and regulatory sciences. By applying "just enough" Linked Data standards, RO-Crate simplifies the process of making research outputs FAIR while also enhancing research reproducibility.

4.1.1 Introduction

The move towards Open Science has increased the need and demand for the publication of artefacts of the research process [Sefton 2021a]. This is particularly apparent in domains that rely on computational experiments; for example, the publication of software, datasets and records of the dependencies that such experiments rely on [Stodden 2016].

It is often argued that the publication of these assets, and specifically software [Lamprecht 2019], workflows [Goble 2020] and data, should follow the FAIR principles [Wilkinson 2016]; namely, that they are Findable, Accessible, Interoperable and Reusable. These principles are agnostic to the *implementation* strategy needed to comply with them. Hence, there has been an increasing amount of work in the development of platforms and specifications that aim to fulfil these goals [Mons 2018].

Important examples include data publication with rich metadata (e.g. Zenodo [Dillen 2019a]), domain-specific data deposition (e.g. PDB [Berman 2007]) and following practices for reproducible research software [Sandve 2013] (e.g. use of containers). While these platforms are useful, experience has shown that it is important to put greater emphasis on the interconnection of the multiple artefacts that make up the research process [Koesten 2021].

The notion of **Research Objects** [Bechhofer 2013] (RO) was introduced to address this connectivity, providing semantically rich *aggregations* of (potentially distributed) resources with a layer of structure over a research study; this is then to be delivered in a *machine-readable format*.

A Research Object combines the ability to bundle multiple types of artefacts together, such as spreadsheets, code, examples, and figures. The RO is augmented with annotations and relationships that describe the artefacts' *context* (e.g. a CSV being used by a script, a figure being a result of a workflow).

This notion of ROs provides a compelling vision as an approach for implementing FAIR data. However, existing Research Object implementations require a large technology stack [Belhajjame 2015], are typically tailored to a particular platform and are also not easily usable by end-users.

To address this gap, a new community came together [Ó Carragáin 2019a] to develop **RO-Crate**—an *approach to package and aggregate research artefacts with their metadata and relationships*. The aim of this paper is to introduce RO-Crate and assess it as a strategy for making multiple types of research artefacts FAIR. Specifically, the contributions of this paper are as follows:

1. An introduction to RO-Crate, its purpose and context.
2. A guide to the RO-Crate community and tooling.
3. Examples of RO-Crate usage, demonstrating its value as connective tissue for different artefacts from different communities.

The rest of this article is organised as follows. We first describe RO-Crate through its development methodology that formed the RO-Crate concept, showing its foundations in Linked Data and emerging principles. We then define RO-Crate technically, before we introduce the community and tooling. We move to analyse RO-Crate with respect to usage in a diverse set of domains. Finally, we present related work and conclude with some remarks including RO-Crate highlights and future work. The appendix to this article (Section 4.3 on page 113) adds a formal definition of RO-Crate using First-Order logic.

4.1.2 RO-Crate

RO-Crate aims to provide an approach to packaging research artefacts with their metadata that can be easily adopted. To illustrate this, let us imagine a research paper reporting on the sequence analysis of proteins obtained from an experiment on mice. The sequence output files, sequence analysis code, resulting data and reports summarising statistical measures are all important and inter-related research artefacts, and consequently would ideally all be co-located in a directory and accompanied with their corresponding metadata. In reality, some of the artefacts (e.g. data or software) will be recorded as external reference to repositories that are not necessarily following the FAIR principles. This conceptual directory, along with the relationships between its constituent digital artefacts, is what the RO-Crate model aims to represent, linking together all the elements of an experiment that are required for the experiment's reproducibility and reusability.

The question then arises as to how the directory with all this material should be packaged in a manner that is accessible and usable by others. This means programmatically and automatically accessible by machines and human-readable. A de facto approach to sharing collections of resources is through compressed archives (e.g. a ZIP file). This solves the problem of "packaging", but it does not guarantee downstream access to all artefacts in a programmatic fashion, nor describe the role of each file in that particular research. Both features, the ability to automatically access and reason about an object, are crucial and lead to the need for explicit metadata about the contents of the folder, describing each and linking them together.

Examples of metadata descriptions across a wide range of domains[3] abound within the literature, both in research data management [Amorim 2016, Farnel 2014, Kurowski 2021] and within library and information systems[4] [Mai Chan 1995, Žumer 2009]. However, many of these approaches require knowledge of metadata schemas, particular annotation systems, or the use of complex software stacks. Indeed, particularly within research, these requirements have led to a lack of adoption and growing frustration with current tooling and specifications [Neylon 2017, Volk 2014, Schriml 2020].

RO-Crate seeks to address this complexity by:

1. Being conceptually simple and easy to understand for developers.
2. Providing strong, easy tooling for integration into community projects.
3. Providing a strong and opinionated guide regarding current best practices.
4. Adopting de-facto standards that are widely used on the Web.

In the following sections we demonstrate how the RO-Crate specification and ecosystem achieve these goals.

4.1.2.1 Development Methodology

It is a good question as to what base level we assume for 'conceptually simple'. We take simplicity to apply at two levels: for the *developers* who produce the platforms and for the *data practitioners* and users of those platforms.

For our development methodology we followed the mantra of working closely with a small group to really get a deep understanding of requirements and ensure rapid feedback loops. We created a pool of early adopter projects from a range of disciplines and groups, primarily addressing developers of platforms. Thus the base level for simplicity was **developer friendliness**.

We assumed a developer familiar with making Web applications with JSON data (who would then learn how to make *RO-Crate JSON-LD*), which informed core design choices for our JSON-level documentation approach and RO-Crate serialization (Section 4.1.2.5 on page 85). Our group of early adopters, growing as the community evolved, drove the RO-Crate requirements and provided feedback through our multiple communication channels including bi-monthly meetings, which we describe in Section 4.1.2.6 on page 88 along with the established norms.

Addressing the simplicity of understanding and engaging with RO-Crate by data practitioners is through the platforms, for example with interactive tools (Section 4.1.3 on page 91) like Describo[5] [La Rosa 2021d] and Jupyter notebooks [Kluyver 2016], and by close discussions

[3] https://rdamsc.bath.ac.uk/scheme-index
[4] https://www.loc.gov/librarians/standards
[5] https://arkisto-platform.github.io/describo/

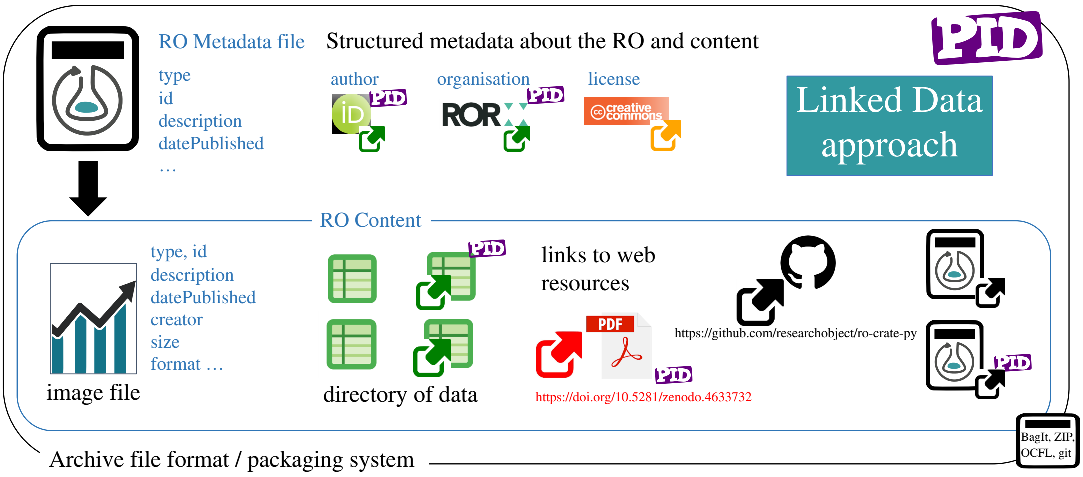


Figure 4.1: Conceptual overview of RO-Crate. A *Persistent Identifier* (PID) [McMurry 2017] points to a *Research Object* (RO), which may be archived using different packaging approaches like BagIt [Kunze 2018], OCFL [OCFL 2020], git or ZIP. The RO is described within a *RO-Crate Metadata File*, providing identifiers for *authors* using ORCID, *organisations* using Research Organization Registry (ROR) [Lammey 2020] and licences such as Creative Commons using SPDX[6] identifiers. The *RO-Crate content* is further described with additional metadata following a Linked Data approach. Data can be embedded files and directories, as well as links to external Web resources, PIDs and nested RO-Crates.

with domain scientists on how to appropriately capture what they determine to be relevant metadata. This ultimately requires a new type of awareness and learning material separate from developer specifications, focusing on the simplicity of extensibility to serve the user needs, along with user-driven development of new RO-Crate Profiles specific for their needs (Section 4.1.4 on page 93).

4.1.2.2 Conceptual Definition

A key premise of RO-Crate is the existence of a wide variety of resources on the Web that can help describe research. As such, RO-Crate relies on the Linked Data principles [Heath 2011]. Figure 4.1 shows the main conceptual elements involved in an RO-Crate: The RO-Crate Metadata File (top) describes the Research Object using structured metadata including external references, coupled with the contained artefacts (bottom) bundled and described by the RO-Crate.

The conceptual notion of a *Research Object* [Bechhofer 2013] is thus realised with the RO-Crate model and serialised using Linked Data constructs within the RO-Crate metadata file.

Linked Data as a foundation The **Linked Data** principles [Bizer 2011] (use of IRIs[7] to identify resources (i.e. artefacts), resolvable via HTTP, enriched with metadata and linked to each other)

[7] IRIs [Dürst 2005] are a generalisation of URIss (which include well-known http/https URLs), permitting international Unicode characters without percent encoding, commonly used on the browser address bar and in HTML5.

are core to RO-Crate; therefore IRIs are used to identify an RO-Crate, its constituent parts and metadata descriptions, and the properties and classes used in the metadata.

RO-Crates are *self-described* and follow the Linked Data principles to describe all of their resources in both human and machine readable manner. Hence, resources are identified using global identifiers (absolute IRIs) where possible; and relationships between two resources are defined with links.

The foundation of Linked Data and shared vocabularies also means that multiple RO-Crates and other Linked Data resources can be indexed, combined, queried, validated or transformed using existing Semantic Web technologies such as SPARQL,[8] SHACL[9] and well established *knowledge graph* triple stores like Apache Jena[10] and OntoText GraphDB.[11]

The possibilities of consuming[12] RO-Crate metadata with such powerful tools gives another strong reason for using Linked Data as a foundation. This use of mature Web[13] technologies also means its developers and consumers are not restricted to the Research Object aspects that have already been specified by the RO-Crate community, but can extend and integrate RO-Crate in multiple standardised ways.

RO-Crate is a self-described container An RO-Crate is defined[14] as a self-described **Root Data Entity** that describes and contains *data entities*, which are further described by referencing *contextual entities*. A **data entity** is either a *file* (i.e. a byte sequence stored on disk somewhere) or a *directory* (i.e. set of named files and other directories). A file does not need to be stored inside the RO-Crate root, it can be referenced via a PID/IRI. A **contextual entity** exists outside the information system (e.g. a Person, a workflow language) and is stored solely by its metadata. The representation of a *data entity* as a byte sequence makes it possible to store a variety of research artefacts including not only data but also, for instance, software and text.

The Root Data Entity is a directory, the *RO-Crate Root*, identified by the presence of the **RO-Crate Metadata File** `ro-crate-metadata.json` (top of Figure 4.1 on the facing page). This file describes the RO-Crate using Linked Data, its content and related metadata using Linked Data in JSON-LD format [Sporny 2014]. This is a W3C standard RDF serialisation that has become popular; it is easy to read by humans while also offering some advantages for data exchange on the Internet. JSON-LD, a subset of the widely supported and well-known JSON format, has tooling available for many programming languages.[15]

[8] https://www.w3.org/TR/sparql11-overview

[9] https://www.w3.org/TR/shacl/

[10] https://jena.apache.org/

[11] https://www.ontotext.com/products/graphdb/

[12] Some consideration is needed in processing of RO-Crates as knowledge graphs, e.g. establishing absolute IRIs for files inside a ZIP archive, detailed in the RO-Crate specification: https://www.researchobject.org/ro-crate/1.1/appendix/relative-uris.html.

[13] Note that an RO-Crate is not required to be published on the Web, see Section 4.1.2.2.

[14] https://www.researchobject.org/ro-crate/1.1/structure.html#ro-crate-metadata-file-ro-crate-metadatajson

[15] https://json-ld.org/#developers

The minimal requirements for the root data entity metadata[16] are `name`, `description` and `datePublished`, as well as a contextual entity identifying its `license`—additional metadata are commonly added to entities depending on the purpose of the particular RO-Crate.

RO-Crates can be stored, transferred or published in multiple ways, e.g. BagIt [Kunze 2018], Oxford Common File Layout [OCFL 2020] (OCFL), downloadable ZIP archives in Zenodo or through dedicated online repositories, as well as published directly on the Web, e.g. using GitHub Pages.[17] Combined with Linked Data identifiers, this caters for a diverse set of storage and access requirements across different scientific domains, from metagenomics workflows producing hundreds of gigabytes of genome data to cultural heritage records with access restrictions for personally identifiable data. Specific *RO-Crate profiles* (Section 4.1.2.4 on page 84) may constrain serialization and publication expectations, and require additional contextual types and properties.

Data Entities are described using Contextual Entities RO-Crate distinguishes between data and contextual entities[18] in a similar way to HTTP terminology's early attempt to separate *information* (data) and *non-information* (contextual) resources [W3C 2007]. Data entities are usually files and directories located by relative IRI references within the RO-Crate Root, but they can also be Web resources or restricted data identified with absolute IRIs, including Persistent Identifiers [McMurry 2017].

As both types of entities are identified by IRIs, their distinction is allowed to be blurry; data entities can be located anywhere and be complex, while contextual entities can have a Web presence beyond their description inside the RO-Crate. For instance `https://orcid.org/0000-0002-1825-0097` is primarily an identifier for a person, but secondarily it is also a Web page and a way to refer to their academic work.

A particular IRI may appear as a contextual entity in one RO-Crate and as a data entity in another; the distinction lies in the fact that data entities can be considered to be *contained* or captured by that RO-Crate (*RO Content* in Figure 4.1 on page 80), while contextual entities mainly *explain* an RO-Crate or its content (although this distinction is not a formal requirement).

In RO-Crate, a referenced contextual entity (e.g. a person identified by ORCID) should always be described within the RO-Crate Metadata File with at least a *type* and *name*, even where their PID might resolve to further Linked Data. This is so that clients are not required to follow every link for presentation purposes, for instance HTML rendering. Similarly any imported extension terms[19] would themselves also have a human-readable description in the case where their PID does not directly resolve to human-readable documentation.

Figure 4.2 on the next page shows a simplified class diagram of RO-Crate, highlighting the different types of data entities and contextual entities that can be aggregated and related. While

[16] https://www.researchobject.org/ro-crate/1.1/root-data-entity.html#direct-properties-of-the-root-data-entity

[17] https://pages.github.com/

[18] https://www.researchobject.org/ro-crate/1.1/contextual-entities.html#contextual-vs-data-entities

[19] https://www.researchobject.org/ro-crate/1.1/appendix/jsonld.html#extending-ro-crate

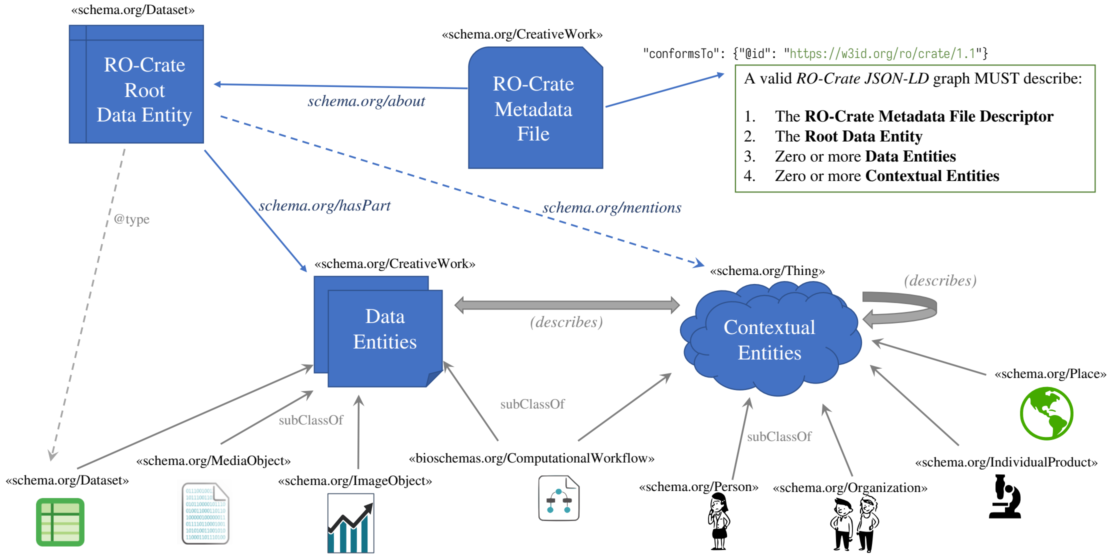

«schema.org/Dataset»
RO-Crate Root Data Entity
«schema.org/CreativeWork»
RO-Crate Metadata File
schema.org/about
"conformsTo": {"@id": "https://w3id.org/ro/crate/1.1"}
A valid RO-Crate JSON-LD graph MUST describe:
1. The RO-Crate Metadata File Descriptor
2. The Root Data Entity
3. Zero or more Data Entities
4. Zero or more Contextual Entities
@type
schema.org/hasPart
schema.org/mentions
«schema.org/CreativeWork»
Data Entities
(describes)
«schema.org/Thing»
Contextual Entities
(describes)
«schema.org/Place»
subClassOf
subClassOf
subClassOf
«schema.org/MediaObject»
«bioschemas.org/ComputationalWorkflow»
«schema.org/IndividualProduct»
«schema.org/Dataset»
«schema.org/ImageObject»
«schema.org/Person»
«schema.org/Organization»

Figure 4.2: Simplified class diagram of RO-Crate. The *RO-Crate Metadata File* conforms to a version of the specification; and contains a JSON-LD graph [Sporny 2014] that describes the entities that make up the RO-Crate. The *RO-Crate Root Data Entity* represent the Research Object as a dataset. The RO-Crate aggregates *data entities* (`hasPart`) which are further described using *contextual entities* (which may include aggregated and non-aggregated data entities). Multiple types and relations from schema.org allow annotations to be more specific, including figures, nested datasets, computational workflows, people, organisations, instruments and places. Contextual entities not otherwise cross-referenced from other entities' properties (*describes*) can be grouped under the root entity (`mentions`).

an RO-Crate would usually contain one or more data entities (`hasPart`), it may also be a pure aggregation of contextual entities (`mentions`).

Guide through Recommended Practices RO-Crate as a specification aims to build a set of recommended practices on how to practically apply existing standards in a common way to describe research outputs and their provenance, without having to learn each of the underlying technologies in detail.

As such, the RO-Crate 1.1 specification[20] [RO-Crate 1.1.3] can be seen as an opinionated and example-driven guide to writing schema.org[21] [Guha 2015] metadata as JSON-LD [Sporny 2014] (see Section 4.1.2.5 on page 85), which leaves it open for implementers to include additional metadata using other schema.org types and properties, or even additional Linked Data vocabularies/ontologies or their own ad-hoc terms.

However the primary purpose of the RO-Crate specification is to assist developers in leveraging Linked Data principles for the focused purpose of describing Research Objects in a structured language, while reducing the steep learning curve otherwise associated with Semantic Web

[20] https://w3id.org/ro/crate/1.1
[21] https://schema.org/

adaptation, like development of ontologies, identifiers, namespaces, and RDF serialization choices.

4.1.2.3 Ensuring Simplicity

One aim of RO-Crate is to be conceptually simple. This simplicity has been repeatedly checked and confirmed through an informal community review process. For instance, in the discussion on supporting ad-hoc vocabularies[22] in RO-Crate, the community explored potential Linked Data solutions. The conventional wisdom in RDF best practices[23] is to establish a vocabulary with a new IRI namespace, formalised using RDF Schema[24] or OWL[25] ontologies. However, this may seem an excessive learning curve for non-experts in semantic knowledge representation, and the RO-Crate community instead agreed on a dual lightweight approach: (i) Document[26] how projects with their own Web-presence can make a pure HTML-based vocabulary, and (ii) provide a community-wide PID namespace under `https://w3id.org/ro/terms` that redirect to simple CSV files maintained in GitHub.[27]

To further verify this idea of simplicity, we have formalised the RO-Crate definition (see *section 4.3 on page 113*). An important result of this exercise is that the underlying data structure of RO-Crate, although conceptually a graph, is represented as a depth-limited tree. This formalisation also emphasises the *boundedness* of the structure; namely, the fact that elements are specifically identified as being either semantically *contained* by the RO-Crate as *Data Entities* (`hasPart`) or mainly referenced (`mentions`) and typed as *external* to the Research Object as *Contextual Entities*. It is worth pointing out that this semantic containment can extend beyond the physical containment of files residing within the RO-Crate Root directory on a given storage system, as the RO-Crate data entities may include any data resource globally identifiable using IRIs.

4.1.2.4 Extensibility and RO-Crate profiles

The RO-Crate specification provides a core set of conventions to describe research outputs using types and properties applicable across scientific domains. However we have found that domain-specific use of RO-Crate will, implicitly or explicitly, form a specialised **profile** of RO-Crate; i.e., *a set of conventions, types and properties that are minimally required and one can expect to be present in that subset of RO-Crates*. For instance, RO-Crates used for exchange of workflows will have to contain a data entity of type `ComputationalWorkflow`, or cultural heritage records should have a `contentLocation`.

Making such profiles explicit allow further reliable programmatic consumption and generation of RO-Crates beyond the core types defined in the RO-Crate specification. Following the RO-Crate mantra of *guidance over strictness*, profiles are mainly *duck-typing* rather than strict syntactic

[22] https://github.com/ResearchObject/ro-crate/issues/71
[23] https://www.w3.org/TR/swbp-vocab-pub/
[24] http://www.w3.org/TR/2014/REC-rdf-schema-20140225/
[25] http://www.w3.org/TR/2012/REC-owl2-overview-20121211/
[26] https://www.researchobject.org/ro-crate/1.1/appendix/jsonld.html#adding-new-or-ad-hoc-vocabulary-terms
[27] https://github.com/ResearchObject/ro-terms

or semantic types, but may also have corresponding machine-readable schemas at multiple levels (file formats, JSON, RDF shapes, RDFS/OWL semantics).

The next version of the RO-Crate specification 1.2 will define a formalization[28] for publishing and declaring conformance to RO-Crate profiles. Such a profile is primarily a human-readable document of before-mentioned expectations and conventions, but may also define a machine-readable profile as a **Profile Crate**: Another RO-Crate that describe the profile and in addition can list schemas for validation, compatible software, applicable repositories, serialization/packaging formats, extension vocabularies, custom JSON-LD contexts and examples (see for example the Workflow RO-Crate profile[29]).

In addition, there are sometimes existing domain-specific metadata formats, but they are either not RDF-based (and thus time-consuming to construct terms for in JSON-LD) or are at a different granularity level that might become overwhelming if represented directly in the RO-Crate Metadata file (e.g. W3C PROV bundle detailing every step execution of a workflow run [Khan 2019]). RO-Crate allows such *alternative metadata files* to co-exist, and be described as data entities with references to the standards and vocabularies they conform to. This simplifies further programmatic consumption even where no filename or file extension conventions have emerged for those metadata formats.

Section 4.1.4 on page 93 examines the observed specializations of RO-Crate use in several domains and their emerging profiles.

4.1.2.5 Technical implementation of the RO-Crate model

The RO-Crate conceptual model has been realised using JSON-LD and schema.org in a prescriptive form as discussed in Section 4.1.2.2 on page 80. These technical choices were made to cater for simplicity from a developer perspective (as introduced in Section 4.1.2.1 on page 79).

JSON-LD[30] [Sporny 2014] provides a way to express Linked Data as a JSON structure, where a *context* provides mapping to RDF properties and classes. While JSON-LD cannot map arbitrary JSON structures to RDF, we found that it does lower the barrier compared to other RDF syntaxes, as the JSON syntax nowadays is a common and popular format for data exchange on the Web.

However, JSON-LD alone has too many degrees of freedom and hidden complexities for software developers to reliably produce and consume without specialised expertise or large RDF software frameworks. A large part of the RO-Crate specification is therefore dedicated to describing the acceptable subset of JSON structures.

[28] https://www.researchobject.org/ro-crate/1.2-DRAFT/profiles
[29] https://w3id.org/workflowhub/workflow-ro-crate/
[30] https://json-ld.org/

RO-Crate JSON-LD RO-Crate mandates[31] the use of flattened, compacted JSON-LD in the RO-Crate Metadata file `ro-crate-metadata.json`[32] where a single `@graph` array contains all the data and contextual entities in a flat list. An example can be seen in the JSON-LD snippet in Listing 4.2 on the next page, describing a simple RO-Crate containing data entities described using contextual entities.

```
{ "@context": "https://w3id.org/ro/crate/1.1/context",
  "@graph": [
    { "@id": "ro-crate-metadata.json",
      "@type": "CreativeWork",
      "conformsTo": {"@id": "https://w3id.org/ro/crate/1.1"},
      "about": {"@id": "./"}
    },
    { "@id": "./",
      "@type": "Dataset",
      "name": "A simplified RO-Crate",
      "author": {"@id": "#alice"},
      "license": {"@id": "https://spdx.org/licenses/CC-BY-4.0"},
      "datePublished": "2021-11-02T16:04:43Z",
      "hasPart": [
        {"@id": "survey-responses-2019.csv"},
        {"@id": "https://example.com/pics/5707039334816454031_o.jpg"}
      ]
    },
    { "@id": "survey-responses-2019.csv",
      "@type": "File",
      "about": {"@id": "https://example.com/pics/5707039334816454031_o.jpg"},
      "author": {"@id": "#alice"}
    },
    { "@id": "https://example.com/pics/5707039334816454031_o.jpg",
      "@type": ["File", "ImageObject"],
      "contentLocation": {"@id": "http://sws.geonames.org/8152662/"},
      "author": {"@id": "https://orcid.org/0000-0002-1825-0097"}
    },
    { "@id": "#alice",
      "@type": "Person",
      "name": "Alice"
    },
    { "@id": "https://orcid.org/0000-0002-1825-0097",
      "@type": "Person",
      "name": "Josiah Carberry"
    },
    { "@id": "http://sws.geonames.org/8152662/",
      "@type": "Place",
      "name": "Catalina Park"
    },
    { "@id": "https://spdx.org/licenses/CC-BY-4.0",
```

[31] https://www.researchobject.org/ro-crate/1.1/appendix/jsonld.html

[32] The avid reader may spot that the RO-Crate Metadata file use the extension `.json` instead of `.jsonld`, this is to emphasise the developer expectations as a JSON format, while the file's JSON-LD nature is secondary. See https://github.com/ResearchObject/ro-crate/issues/82.

```
        "@type": "CreativeWork",
        "name": "Creative Commons Attribution 4.0"
      }
    ]
  }
```

Listing 4.2: Simplified[33]RO-Crate metadata file showing the flattened compacted JSON-LD `@graph` array containing the data entities and contextual entities, cross-referenced using `@id`. The `ro-crate-metadata.json` entity self-declares conformance with the RO-Crate specification using a versioned persistent identifier, further RO-Crate descriptions are on the root data entity `./` or any of the referenced data or contextual entities. This is exemplified by the data entity `ImageObject` referencing contextual entities for `contentLocation` and `author` that differs from that of the overall RO-Crate. In this crate, `about` of the CSV data entity reference the `ImageObject`, which then take the roles of both a data entity and contextual entity. While `Person` entities ideally are identified with ORCID PIDs as for Josiah, `#alice` is here in contrast an RO-Crate local identifier, highlighting the pragmatic "just enough" Linked Data approach.

In this flattened profile of JSON-LD, each `{entity}` are directly under `@graph` and represents the RDF triples with a common *subject* (`@id`), mapped *properties* like `hasPart`, and *objects*—as either literal `"string"` values, referenced `{objects}` (which properties are listed in its own entity), or a JSON `[list]` of these. If processed as JSON-LD, this forms an RDF graph by matching the `@id` IRIs and applying the `@context` mapping to schema.org terms.

Flattened JSON-LD When JSON-LD 1.0 [Sporny 2014] was proposed, one of the motivations was to seamlessly apply an RDF nature on top of regular JSON as frequently used by Web APIs. JSON objects in APIs are frequently nested with objects at multiple levels, and the perhaps most common form of JSON-LD is the compacted form[34] which follows this expectation (JSON-LD 1.1[35] further expands these capabilities, e.g. allowing nested `@context` definitions).

While this feature of JSON-LD can be seen as a way to "hide" its RDF nature, we found that the use of nested trees (e.g. a `Person` entity appearing as `author` of a `File` which nests under a `Dataset` with `hasPart`) counter-intuitively forces consumers to consider the JSON-LD as an RDF Graph, since an identified `Person` entity can appear at multiple and repeated points of the tree (e.g. author of multiple files), necessitating node merging or duplication, which can become complicated as this approach also invites the use of *blank nodes* (entities missing `@id`).

By comparison, a single flat `@graph` array approach, as required by RO-Crate, means that applications can choose to process and edit each entity as pure JSON by a simple lookup based on `@id`. At the same time, lifting all entities to the same level reflects the Research Object principles [Bechhofer 2013] in that describing the context and provenance is just as important

[33] Recommended properties for types shown in listing also include `affiliation`, `citation`, `contactPoint`, `description`, `encodingFormat`, `funder`, `geo`, `identifier`, `keywords`, `publisher`; these properties and corresponding contextual entities are excluded here for brevity. See complete example https://www.researchobject.org/2021-packaging-research-artefacts-with-ro-crate/listing1/.

[34] https://json-ld.org/spec/REC/json-ld/20140116/#compacted-document-form

[35] https://www.w3.org/TR/2020/REC-json-ld11-20200716/

as describing the data, and the requirement of `@id` of every entity forces RO-Crate generators to consciously consider existing IRIs and identifiers.[36]

JSON-LD context In JSON-LD, the `@context` is a reference to another JSON-LD document that provides mapping from JSON keys to Linked Data term IRIs, and can enable various JSON-LD directives to cater for customised JSON structures for translating to RDF.

RO-Crate reuses vocabulary terms and IRIs from schema.org, but provides its own versioned JSON-LD context,[37] which has a flat list with the mapping from JSON-LD keys to their IRI equivalents (e.g. key `"author"` maps to the http://schema.org/author property).

The rationale behind this decision is to support JSON-based RO-Crate applications that are largely unaware of JSON-LD, that still may want to process the `@context` to find or add Linked Data definitions of otherwise unknown properties and types. Not reusing the official schema.org context means RO-Crate is also able to map in additional vocabularies where needed, namely the *Portland Common Data Model* (PCDM) [Cossu 2018] for repositories and Bioschemas [Gray 2017] for describing computational workflows. RO-Crate profiles may extend[38] the `@context` to re-use additional domain-specific ontologies.

Similarly, while the schema.org context currently[39] have `"@type": "@id"` annotations for implicit object properties, RO-Crate JSON-LD distinguishes explicitly between references to other entities `{"@id": "#alice"}` and string values `"Alice"`—meaning RO-Crate applications can find references for corresponding entities and IRIs without parsing the `@context` to understand a particular property. Notably this is exploited by the *ro-crate-html-js* [ro-crate-html-js] tool to provide reliable HTML rendering for otherwise unknown properties and types.

4.1.2.6 RO-Crate Community

The RO-Crate conceptual model, implementation and best practices are developed by a growing community of researchers, developers and publishers. RO-Crate's community is a key aspect of its effectiveness in making research artefacts FAIR. Fundamentally, the community provides the overall context of the implementation and model and ensures its interoperability.

The RO-Crate community consists of:

1. A diverse set of people representing a variety of stakeholders.
2. A set of collective norms.
3. An open platform that facilitates communication (GitHub, Google Docs, monthly teleconferences).

[36] https://www.researchobject.org/ro-crate/1.1/appendix/jsonld.html#describing-entities-in-json-ld
[37] https://w3id.org/ro/crate/1.1/context
[38] https://www.researchobject.org/ro-crate/1.1/appendix/jsonld.html#extending-ro-crate
[39] https://schema.org/version/13.0/schemaorg-current-http.jsonld

People The initial concept of RO-Crate was formed at the first Workshop on Research Objects[40] (RO2018), held as part of the IEEE conference on eScience. This workshop followed up on considerations made at a Research Data Alliance (RDA) meeting on Research Data Packaging[41] that found similar goals across multiple data packaging efforts [Ó Carragáin 2019a]: simplicity, structured metadata and the use of JSON-LD.

An important outcome of discussions that took place at RO2018 was the conclusion that the original Wf4Ever Research Object ontologies [Belhajjame 2015], in principle sufficient for packaging research artefacts with rich descriptions, were, in practice, considered inaccessible for regular programmers (e.g., Web developers) and in danger of being incomprehensible for domain scientists due to their reliance on Semantic Web technologies and other ontologies.

DataCrate [Sefton 2018] was presented at RO2018 as a promising lightweight alternative approach, and an agreement was made by a group of volunteers to attempt building what was initially called *"RO Lite"* as a combination of DataCrate's implementation and Research Object's principles.

This group, originally made up of library and Semantic Web experts, has subsequently grown to include domain scientists, developers, publishers and more. This perspective of multiple views led to the specification being used in a variety of domains, from bioinformatics and regulatory submissions to humanities and cultural heritage preservation.

The RO-Crate community is strongly engaged with the European-wide biology/bioinformatics collaborative e-Infrastructure ELIXIR [Crosswell 2012], along with European Open Science Cloud[42] (EOSC) projects including EOSC-Life,[43] FAIRplus,[44] CS3MESH4EOSC[45] and BY-COVID.[46] RO-Crate has also established collaborations with Bioschemas [Gray 2017], GA4GH [Rehm 2021], OpenAIRE [Rettberg 2015] and multiple H2020 projects.

A key set of stakeholders are developers: the RO-Crate community has made a point of attracting developers who can implement the specifications but, importantly, keeps "developer user experience" in mind. This means that the specifications are straightforward to implement and thus do not require expertise in technologies that are not widely deployed.

This notion of catering to "developer user experience" is an example of the set of norms that have developed and now define the community.

Norms The RO-Crate community is driven by informal conventions and notions that are prevalent but not neccessarily written down. Here, we distil what we as authors believe are the critical set of norms that have facilitated the development of RO-Crate and contributed to the

[40] https://www.researchobject.org/ro2018/
[41] https://rd-alliance.org/approaches-research-data-packaging-rda-11th-plenary-bof-meeting
[42] https://eosc.eu/
[43] https://www.eosc-life.eu/
[44] https://fairplus-project.eu/
[45] https://cs3mesh4eosc.eu/
[46] https://by-covid.eu/

ability for RO-Crate research packages to be FAIR. This is not to say that there are no other norms within the community nor that everyone in the community holds these uniformly. Instead, what we emphasise is that these norms are helpful and also shaped by community practices:

1. Simplicity.
2. Developer friendliness.
3. Focus on examples and best practices rather than rigorous specification.
4. Reuse "just enough" Web standards.

A core norm of RO-Crate is that of **simplicity**, which sets the scene for how we guide developers to structure metadata with RO-Crate. We focus mainly on documenting simple approaches to the most common use cases, such as authors having an affiliation. This norm also influences our take on **developer friendliness**; for instance, we are using the Web-native JSON format, allowing only a few of JSON-LD's flexible Linked Data features. Moreover, the RO-Crate documentation is largely built up by **examples** showcasing **best practices**, rather than rigorous specifications. We build on existing **Web standards** that themselves are defined rigorously, which we utilise *"**just enough**"* in order to benefit from the advantages of Linked Data (e.g., extensions by namespaced vocabularies), without imposing too many developer choices or uncertainties (e.g., having to choose between the many RDF syntaxes).

While the above norms alone could easily lead to the creation of "yet another" JSON format, we keep the goal of **FAIR interoperability** of the captured metadata, and therefore follow closely FAIR best practices and current developments such as data citations, PIDs, open repositories and recommendations for sharing research outputs and software.

Open Platforms The critical infrastructure that enables the community around RO-Crate is the use of open development platforms. This underpins the importance of open community access to supporting FAIR. Specifically, it is difficult to build and consume FAIR research artefacts without being able to access the specifications, understand how they are developed, know about any potential implementation issues, and discuss usage to evolve best practices.

The development of RO-Crate was driven by capturing documentation of real-life examples and best practices rather than creating a rigorous specification. At the same time, we agreed to be opinionated on the syntactic form to reduce the jungle of implementation choices; we wanted to keep the important aspects of Linked Data to adhere to the FAIR principles while retaining the option of combining and extending the structured metadata using the existing Semantic Web stack, not just build a standalone JSON format.

Further work during 2019 started adapting the DataCrate documentation through a more collaborative and exploratory *RO Lite* phase, initially using Google Docs for review and discussion, then moving to GitHub as a collaboration space for developing what is now the RO-Crate specification, maintained[47] as Markdown in GitHub Pages and published through Zenodo.

[47] https://github.com/researchobject/ro-crate/

In addition to the typical Open Source-style development with GitHub issues and pull requests, the RO-Crate Community have, at time of writing, two regular monthly calls, a Slack channel and a mailing list for coordinating the project; also many of its participants collaborate on RO-Crate at multiple conferences and coding events such as the ELIXIR BioHackathon.[48] The community is jointly developing the RO-Crate specification and Open Source tools, as well as providing support and considering new use cases. The RO-Crate Community[49] is open for anyone to join, to equally participate under a code of conduct, and as of October 2021 has more than 50 members (see A.2.1 on page 213).

4.1.3 RO-Crate Tooling

The work of the community has led to the development of a number of tools for creating and using RO-Crates. Table 4.1 on page 93 shows the current set of implementations[50]. Reviewing this list, one can see support for commonly used programming languages, including Python, JavaScript, and Ruby. Additionally, the tools can be integrated into commonly used research environments, in particular, the command line tool *ro-crate-html-js* [ro-crate-html-js] for creating a human-readable preview of an RO-Crate as a sidecar HTML file. Furthermore, there are tools that cater to end-users (*Describo* [La Rosa 2021d], *WorkflowHub* [WorkflowHub 2023]), in order to simplify creating and managing RO-Crate. For example, Describo was developed to help researchers of the Australian Criminal Characters project[51] to annotate historical prisoner records for greater insight into the history of Australia [Piper 2020].

While the development of these tools is promising, our analysis of their maturity status shows that the majority of them are in the Beta stage. This is partly due to the fact that the RO-Crate specification itself only recently reached 1.0 status, in November 2019 [RO-Crate 1.0]. Now that there is a fixed point of reference: With version 1.1 (October 2020) [RO-Crate 1.1] RO-Crate has stabilised based on feedback from application development, and now we are seeing a further increase in the maturity of these tools, along with the creation of new ones.

Given the stage of the specification, these tools have been primarily targeting developers, essentially providing them with the core libraries for working with RO-Crate. Another target has been that of research data managers who need to manage and curate large amounts of data.

[48] https://biohackathon-europe.org/

[49] https://www.researchobject.org/ro-crate/community

[50] Several new implementations have appeared since the publication of this article, see Section 6.1.3 on page 194.

[51] https://criminalcharacters.com/

Table 4.1: Applications and libraries implementing RO-Crate, targeting different types of users across multiple programming languages. Status is indicative as assessed by this work (Alpha < Beta < Release Candidate (RC) < Release).

Tool Name	Targets	Language /Platform	Status	Brief Description
Describo [La Rosa 2021d]	Research Data Managers	NodeJS (Desktop)	RC	Interactive desktop application to create, update and export RO-Crates for different profiles
Describo Online [La Rosa 2021c]	Platform developers	NodeJS (Web)	Alpha	Web-based application to create RO-Crates using cloud storage
ro-crate-excel [Lynch 2022]	Data managers	JavaScript	Beta	Command-line tool to create/edit RO-Crates with spreadsheets
ro-crate-html-js [ro-crate-html-js]	Developers	JavaScript	Beta	HTML rendering of RO-Crate
ro-crate-js [Sefton 2021b]	Research Data Managers	JavaScript	Alpha	Library for creating/manipulating crates; basic validation code
ro-crate-ruby [Bacall 2022b]	Developers	Ruby	Beta	Ruby library for reading/writing RO-Crate, with workflow support
ro-crate-py [De Geest 2023a])	Developers	Python	Alpha	Object-oriented Python library for reading/writing RO-Crate and use by Jupyter Notebook
WorkflowHub [WorkflowHub 2023]	Workflow users	Ruby	Beta	Workflow repository; imports and exports Workflow RO-Crate
Life Monitor [CRS4 2022]	Workflow developers	Python	Alpha	Workflow testing and monitoring service; Workflow Testing profile of RO-Crate
SCHeMa [Vergoulis 2022]	Workflow users	PHP	Alpha	Workflow execution using RO-Crate as exchange mechanism [Vergoulis 2021]
galaxy2cwl [Eguinoa 2020]	Workflow developers	Python	Alpha	Wraps Galaxy workflow as Workflow RO-Crate
Modern PARADISEC [La Rosa 2021a]	Repository managers	Platform	Beta	Cultural Heritage portal based on OCFL and RO-Crate
ONI express [Arkisto 2022]	Repository managers	Platform	Beta	Platform for publishing data and documents stored in an OCFL repository via a Web interface

Table 4.1: Applications and libraries implementing RO-Crate, targeting different types of users across multiple programming languages. Status is indicative as assessed by this work (Alpha < Beta < Release Candidate (RC) < Release).

Tool Name	Targets	Language /Platform	Status	Brief Description
ocfl-tools [La Rosa 2021b]	Developers	JavaScript (CLI)	Beta	Tools for managing RO-Crates in an OCFL repository
RO Composer [Bacall 2019]	Repository developers	Java	Alpha	REST API for gradually building ROs for given profile.
RDA maDMP Mapper [Arfaoui 2020]	Data Management Plan users	Python	Beta	Mapping between machine-actionable data management plans (maDMP) and RO-Crate [Miksa 2020]
Ro-Crate_2_ma-DMP [Brenner 2020]	Data Management Plan users	Python	Beta	Convert between machine-actionable data management plans (maDMP) and RO-Crate
CheckMyCrate [Belchev 2021]	Developers	Python (CLI)	Alpha	Validation according to Workflow RO-Crate profile
RO-Crates-and-Excel [Zoubek 2021]	Data Managers	Java (CLI)	Alpha	Describe column/data details of spreadsheets as RO-Crate using DataCube vocabulary

4.1.4 Profiles of RO-Crate in use

RO-Crate fundamentally forms part of an infrastructure to help build FAIR research artefacts. In other words, the key question is whether RO-Crate can be used to share and (re)use research artefacts. Here we look at three research domains where RO-Crate is being applied: Bioinformatics, Regulatory Science and Cultural Heritage. In addition, we note how RO-Crate may have an important role as part of machine-actionable data management plans and institutional repositories.

From these varied uses of RO-Crate we observe natural differences in their detail level and the type of entities described by the RO-Crate. For instance, on submission of an RO-Crate to a workflow repository, it is reasonable to expect the RO-Crate to contain at least one workflow, ideally with a declared licence and workflow language. Specific additional recommendations such as on identifiers is also needed to meet the emerging requirements of FAIR Digital Objects.[52] Work has now begun[53] to formalise these different *profiles* of RO-Crates, which may impose additional constraints based on the needs of a specific domain or use case.

[52] https://fairdo.org/

[53] https://github.com/ResearchObject/ro-crate/issues/153 was implemented after publication of this article—see

4.1.4.1 Bioinformatics workflows

WorkflowHub.eu[54] is a European cross-domain registry of computational workflows, supported by European Open Science Cloud projects, e.g. EOSC-Life,[55] and research infrastructures including the pan-European bioinformatics network ELIXIR[56] [Crosswell 2012]. As part of promoting workflows as reusable tools, WorkflowHub includes documentation and high-level rendering of the workflow structure independent of its native workflow definition format. The rationale is that a domain scientist can browse all relevant workflows for their domain, before narrowing down their workflow engine requirements. As such, the WorkflowHub is intended largely as a registry of workflows already deposited in repositories specific to particular workflow languages and domains, such as UseGalaxy.eu [Baker 2020] and Nextflow nf-core [Ewels 2020].

We here describe three different RO-Crate profiles developed for use with WorkflowHub.

Profile for describing workflows Being cross-domain, WorkflowHub has to cater for many different workflow systems. Many of these, for instance Nextflow [Di Tommaso 2017] and Snakemake [Köster 2012], by virtue of their script-like nature, reference multiple neighbouring files typically maintained in a GitHub repository. This calls for a data exchange method that allows keeping related files together. WorkflowHub has tackled this problem by adopting RO-Crate as the packaging mechanism [Bietrix 2021], typing and annotating the constituent files of a workflow and—crucially—marking up the workflow language, as many workflow engines use commonfile extensions like `*.xml` and `*.json`. Workflows are further described with authors, license, diagram previews and a listing of their inputs and outputs. RO-Crates can thus be used for interoperable deposition of workflows to WorkflowHub, but are also used as an archive for downloading workflows, embedding metadata registered with the WorkflowHub entry and translated workflow files such as abstract Common Workflow Language (CWL) [Crusoe 2022] definitions and diagrams [Goble 2021].

RO-Crate acts therefore as an interoperability layer between registries, repositories and users in WorkflowHub. The iterative development between WorkflowHub developers and the RO-Crate community heavily informed the creation of the Bioschemas [Gray 2017] profile for Computational Workflows [57], which again informed the RO-Crate 1.1 specification on workflows [58] and led to the RO-Crate Python library [De Geest 2023a] and WorkflowHub's **Workflow RO-Crate profile**[59], which, in a similar fashion to RO-Crate itself, recommends which workflow resources and descriptions are required. This co-development across project boundaries exemplifies the drive for simplicity and for establishing best practices.

Section 6.1.2.4 on page 191.

[54] https://workflowhub.eu/

[55] https://www.eosc-life.eu/

[56] https://elixir-europe.org/

[57] https://bioschemas.org/profiles/ComputationalWorkflow/1.0-RELEASE/

[58] https://www.researchobject.org/ro-crate/1.1/workflows.html

[59] https://w3id.org/workflowhub/workflow-ro-crate/1.0

Profile for recording workflow runs RO-Crates in WorkflowHub have so far been focused on workflows that are ready to be run, and development of WorkflowHub is now creating a **Workflow Run RO-Crate profile**[60] for the purposes of benchmarking, testing and executing workflows. As such, RO-Crate serves as a container of both a *workflow definition* that may be executed and of a particular *workflow execution with test results*.

This workflow run profile is a continuation of our previous work with capturing workflow provenance in a Research Object in CWLProv [Khan 2019] and TavernaPROV [Soiland-Reyes 2016]. In both cases, we used the PROV Ontology [Lebo 2013a], including details of every task execution with all the intermediate data, which required significant workflow engine integration.[61]

Simplifying from the CWLProv approach, the planned Workflow Run RO-Crate profile will use a high level schema.org provenance[62] for the input/output boundary of the overall workflow execution. This *Level 1 workflow provenance* [Khan 2019] can be expressed generally across workflow languages with minimal workflow engine changes, with the option of more detailed provenance traces as separate PROV artefacts in the RO-Crate as data entities. In the current development of the Specimen Data Refinery (SDR)[63] [Walton 2020a], these RO-Crates will[64] document the text recognition workflow runs of digitised biological specimens, exposed as FAIR Digital Objects [De Smedt 2020].

WorkflowHub has recently enabled minting of Digital Object Identifiers (DOIs), a PID commonly used for scholarly artefacts, for registered workflows, e.g. `10.48546/workflowhub.workflow.56.1` [Lowe 2021b], lowering the barrier for citing workflows as computational methods along with their FAIR metadata—captured within an RO-Crate. While it is not an aim for WorkflowHub to be a repository of workflow runs and their data, RO-Crates of *exemplar workflow runs* serve as useful workflow documentation, as well as being an exchange mechanism that preserves FAIR metadata in a diverse workflow execution environment.

Profile for testing workflows The value of computational workflows, however, is potentially undermined by the "collapse" over time of the software and services they depend upon: for instance, software dependencies can change in a non-backwards-compatible manner, or active maintenance may cease; an external resource, such as a reference index or a database query service, could shift to a different URL or modify its access protocol; or the workflow itself may develop hard-to-find bugs as it is updated. This *workflow decay* can take a big toll on the workflow's reusability and on the reproducibility of any processes it evokes [Zhao 2012].

For this reason, WorkflowHub is complemented by a monitoring and testing service called LifeMonitor [CRS4 2022], also supported by EOSC-Life. LifeMonitor's main goal is to assist in the creation, periodic execution and monitoring of workflow tests, enabling the early detection of

[60] Section 5.4 on page 152

[61] CWLProv and TavernaProv predate RO-Crate, but use RO-Bundle [Soiland-Reyes 2014], a similar Research Object packaging method with JSON-LD metadata.

[62] https://www.researchobject.org/ro-crate/1.1/provenance.html#software-used-to-create-files

[63] https://github.com/DiSSCo/SDR

[64] See Sections 5.2 on page 133 and 5.3 on page 148

software collapse in order to minimise its detrimental effects. The communication of metadata related to workflow testing is achieved through the adoption of a **Workflow Testing RO-Crate profile**[65] stacked on top of the *Workflow RO-Crate* profile. This further specialisation of Workflow RO-Crate allows to specify additional testing-related entities (test suites, instances, services, etc.), leveraging RO-Crate's extension mechanism[66] through the addition of terms from custom namespaces.

In addition to showcasing RO-Crate's extensibility, the testing profile is an example of the format's flexibility and adaptability to the different needs of the research community. Though ultimately related to a computational workflow, in fact, most of the testing-specific entities are more about describing a protocol for interacting with a monitoring service than a set of research outputs and its associated metadata. Indeed, one of LifeMonitor's main functionalities is monitoring and reporting on test suites running on existing Continuous Integration (CI) services, which is described in terms of service URLs and job identifiers in the testing profile. In principle, in this context, data could disappear altogether, leading to an RO-Crate consisting entirely of contextual entities. Such an RO-Crate acts more as an exchange format for communication between services (WorkflowHub and LifeMonitor) than as an aggregator for research data and metadata, providing a good example of the format's high versatility.

4.1.4.2 Regulatory Sciences

BioCompute Objects[67] (BCO) [Alterovitz 2018] is a community-led effort to standardise submissions of computational workflows to biomedical regulators. For instance, a genomics sequencing pipeline, as part of a personalised cancer treatment study, can be submitted to the US Food and Drugs Administration (FDA) for approval. BCOs are formalised in the standard IEEE 2791-2020 [IEEE 2791-2020] as a combination of JSON Schemas[68] that define the structure of JSON metadata files describing exemplar workflow runs in detail, covering aspects such as the usability and error domain of the workflow, its runtime requirements, the reference datasets used and representative output data produced.

BCOs provide a structured view over a particular workflow, informing regulators about its workings independently of the underlying workflow definition language. However, BCOs have only limited support for additional metadata.[69] For instance, while the BCO itself can indicate authors and contributors, and in particular regulators and their review decisions, it cannot describe the provenance of individual data files or workflow definitions.

As a custom JSON format, BCOs cannot be extended with Linked Data concepts, except by adding an additional top-level JSON object formalised in another JSON Schema. A BCO and workflow submitted by upload to a regulator will also frequently consist of multiple cross-related

[65] https://lifemonitor.eu/workflow_testing_ro_crate

[66] https://www.researchobject.org/ro-crate/1.1/appendix/jsonld.html#extending-ro-crate

[67] https://biocomputeobject.org/

[68] https://w3id.org/ieee/ieee-2791-schema/

[69] IEEE 2791-2020 do permit user extensions in the *extension domain* by referencing additional JSON Schemas.

files. Crucially, there is no way to tell whether a given `*.json` file is a BCO file, except by reading its content and check for its `spec_version`.

We can then consider how a BCO and its referenced artefacts can be packaged and transferred following FAIR principles. **BCO RO-Crate**[70] [Soiland-Reyes 2021], part of the BioCompute Object user guides, defines a set of best practices for wrapping a BCO with a workflow, together with its exemplar outputs in an RO-Crate, which then provides typing and additional provenance metadata of the individual files, workflow definition, referenced data and the BCO metadata itself.

Here the BCO is responsible for describing the *purpose* of a workflow and its run at an abstraction level suitable for a domain scientist, while the more open-ended RO-Crate describes the surroundings of the workflow, classifying and relating its resources and providing provenance of their existence beyond the BCO. This emerging *separation of concerns* is shown in Figure 4.3 on the next page, and highlights how RO-Crate is used side-by-side of existing standards and tooling, even where there are apparent partial overlaps.

A similar separation of concerns can be found if considering the RO-Crate as a set of files, where the *transport-level* metadata, such as checksum of files, are delegated to separate BagIt[71] manifests, a standard focusing on the preservation challenges of digital libraries [Kunze 2018]. As such, RO-Crate metadata files are not required to iterate all the files in their folder hierarchy, only those that benefit from being described.

Specifically, a BCO description alone is insufficient for reliable re-execution of a workflow, which would need a compatible workflow engine depending on the original workflow definition language, so IEEE 2791 recommends using Docker[72] or Conda.[73] Thus, we can consider BCO RO-Crate as a stack: transport-level manifests of files (BagIt), provenance, typing and context of those files (RO-Crate), workflow overview and purpose (BCO), interoperable workflow definition (CWL) and tool distribution (Docker).

4.1.4.3 Digital Humanities: Cultural Heritage

The Pacific And Regional Archive for Digital Sources in Endangered Cultures (PARADISEC[74]) [Thieberger 2012] maintains a repository of more than 500,000 files documenting endangered languages across more than 16,000 items, collected and digitised over many years by researchers interviewing and recording native speakers across the region.

The Modern PARADISEC demonstrator[75] has been proposed[76] as an update to the 18 year old infrastructure, to also help long-term preservation of these artefacts in their digital form. The

[70] https://biocompute-objects.github.io/bco-ro-crate/

[71] https://www.researchobject.org/ro-crate/1.1/appendix/implementation-notes.html#adding-ro-crate-to-bagit

[72] https://www.docker.com/

[73] https://docs.conda.io/

[74] https://www.paradisec.org.au/

[75] https://mod.paradisec.org.au/

[76] https://arkisto-platform.github.io/case-studies/paradisec/

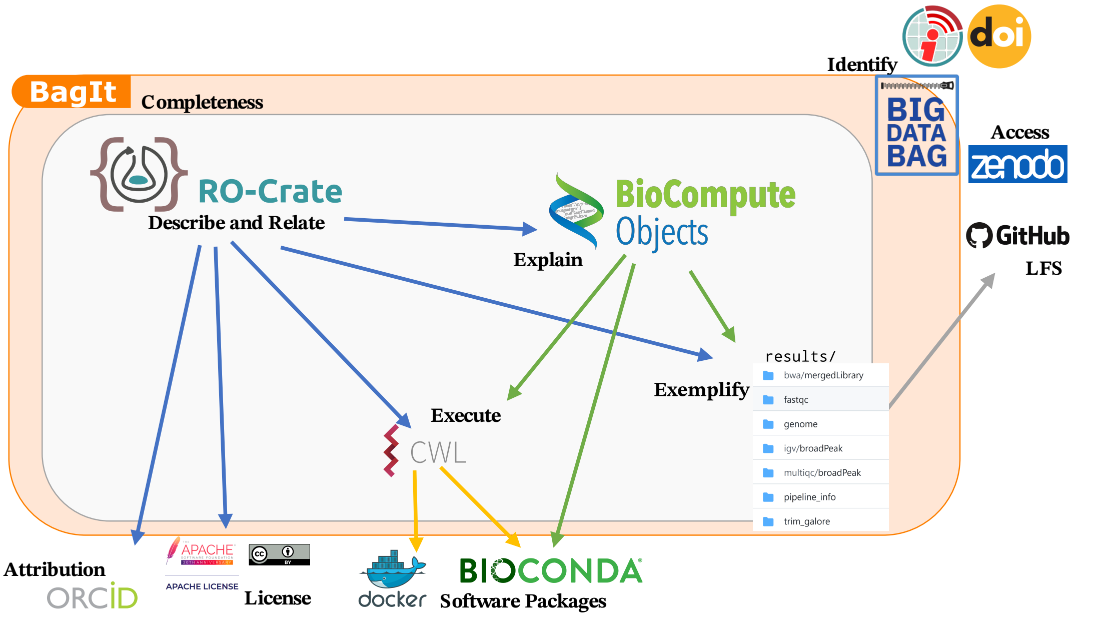


Figure 4.3: Separation of Concerns in BCO RO-Crate. BioCompute Object (IEEE2791) is a JSON file that structurally explains the purpose and implementation of a computational workflow, for instance implemented in Common Workflow Language (CWL), that installs the workflow's software dependencies as Docker containers or BioConda packages. An example execution of the workflow shows the different kinds of result outputs, which may be external, using GitHub LFS [GitHub 2021] to support larger data. RO-Crate gathers all these local and external resources, relating them and giving individual descriptions, for instance permanent DOI identifiers for reused datasets accessed from Zenodo, but also adding external identifiers to attribute authors using ORCID or to identify which licences apply to individual resources. The RO-Crate and its local files are captured in a BagIt whose checksum ensures completeness, combined with Big Data Bag [Chard 2016] features to "complete" the bag with large external files such as the workflow outputs.

demonstrator uses RO-Crate to describe the overall structure and to capture the metadata of each item. The existing PARADISEC data collection has been ported and captured as RO-Crates. A Web portal then exposes the repository and its entries by indexing the RO-Crate metadata files, presenting a domain-specific view of the items—the RO-Crate is "hidden" and does not change the user interface.

The PARADISEC use case takes advantage of several RO-Crate features and principles. Firstly, the transcribed metadata are now independent of the PARADISEC platform and can be archived, preserved and processed in its own right, using schema.org as base vocabulary and extended with PARADISEC-specific terms.

In this approach, RO-Crate is the holder of itemised metadata, stored in regular files that are organised using Oxford Common File Layout[77] (OCFL) [OCFL 2020], which ensures file integrity and versioning on a regular shared file system. This lightweight infrastructure also gives flexibility for future developments and maintenance. For example a consumer can use Linked Data software such as a graph database and query the whole corpora using SPARQL triple patterns across multiple RO-Crates. For long term digital preservation, beyond the lifetime of PARADISEC portals, a "last resort" fallback is storing the generic RO-Crate HTML preview [ro-crate-html-js]. Such human-readable rendering of RO-Crates can be hosted as static files by any Web server, in line with the approach taken by the Endings Project.[78]

4.1.4.4 Machine-actionable Data Management Plans

Machine-actionable Data Management Plans (maDMPs) have been proposed as an improvement to automate FAIR data management tasks in research [Miksa 2019b]; maDMPs use PIDs and controlled vocabularies to describe what happens to data over the research life cycle [Cardoso 2020a]. The Research Data Alliance's *DMP Common Standard* for maDMPs [Miksa 2019a] is one such formalisation for expressing maDMPs, which can be expressed as Linked Data using the DMP Common Standard Ontology [Cardoso 2020b], a specialisation of the W3C Data Catalog Vocabulary (DCAT) [Albertoni 2020]. RDA maDMPs are usually expressed using regular JSON, conforming to the DMP JSON Schema.

A mapping has been produced between Research Object Crates and Machine-actionable Data Management Plans [Miksa 2020], implemented by the RO-Crate RDA maDMP Mapper [Arfaoui 2020]. A similar mapping has been implemented by *RO-Crate_2_ma-DMP* [Brenner 2020]. In both cases, a maDMP can be converted to a RO-Crate, or vice versa. In [Miksa 2020] this functionality caters for two use cases:

1. Start a skeleton data management plan based on an existing RO-Crate dataset, e.g. an RO-Crate from WorkflowHub.

[77] https://ocfl.io/1.0/spec/

[78] The Endings Project https://endings.uvic.ca/ is a five-year project funded by the Social Sciences and Humanities Research Council (SSHRC) that is creating tools, principles, policies and recommendations for digital scholarship practitioners to create accessible, stable, long-lasting resources in the humanities.

2. Instantiate an RO-Crate based on a data management plan.

An important nuance here is that data management plans are (ideally) written in *advance* of data production, while RO-Crates are typically created to describe data *after* it has been generated. What is significant to note in this approach is the importance of **templating** in order to make both tasks automatable and achievable, and how RO-Crate can fit into earlier stages of the research life cycle.

4.1.4.5 Institutional data repositories—Harvard Data Commons

The concept of a **Data Commons** for research collaboration was originally defined as *"cyberinfrastructure that co-locates data, storage, and computing infrastructure with commonly used tools for analysing and sharing data to create an interoperable resource for the research community"* [Grossman 2016]. More recently, Data Commons has been established to mean integration of active data-intensive research with data management and archival best practices, along with a supporting computational infrastructure. Furthermore, the Commons features tools and services, such as computation clusters and storage for scalability, data repositories for disseminating and preserving regular, but also large or sensitive datasets, and other research assets. Multiple initiatives were undertaken to create Data Commons on national, research, and institutional levels. For example, the Australian Research Data Commons (ARDC)[79] [Barker 2019] is a national initiative that enables local researchers and industries to access computing infrastructure, training, and curated datasets for data-intensive research. NCI's Genomic Data Commons[80] (GDC) [Jensen 2017] provides the cancer research community with access to a vast volume of genomic and clinical data. Initiatives such as Research Data Alliance (RDA) Global Open Research Commons[81] propose standards for the implementation of Data Commons to prevent them becoming "data silos" and thus, enable interoperability from one Data Commons to another.

Harvard Data Commons [Crosas 2020] aims to address the challenges of data access and cross-disciplinary research within a research institution. It brings together multiple institutional schools, libraries, computing centres and the Harvard Dataverse[82] data repository. Dataverse[83] [Crosas 2011] is a free and open-source software platform to archive, share and cite research data. The Harvard Dataverse repository is the largest of 70 Dataverse installations worldwide, containing over 120K datasets with about 1.3M data files (as of 2021-11-16). Working toward the goal of facilitating collaboration and data discoverability and management within the university, Harvard Data Commons has the following primary objectives:

1. The integration of Harvard Research Computing with Harvard Dataverse by leveraging Globus endpoints [Chard 2014]; this will allow an automatic transfer of large datasets to the repository. In some cases, only the metadata will be transferred while the data stays

[79] https://ardc.edu.au/

[80] https://gdc.cancer.gov/

[81] https://www.rd-alliance.org/groups/global-open-research-commons-ig

[82] https://dataverse.harvard.edu/

[83] https://dataverse.org/

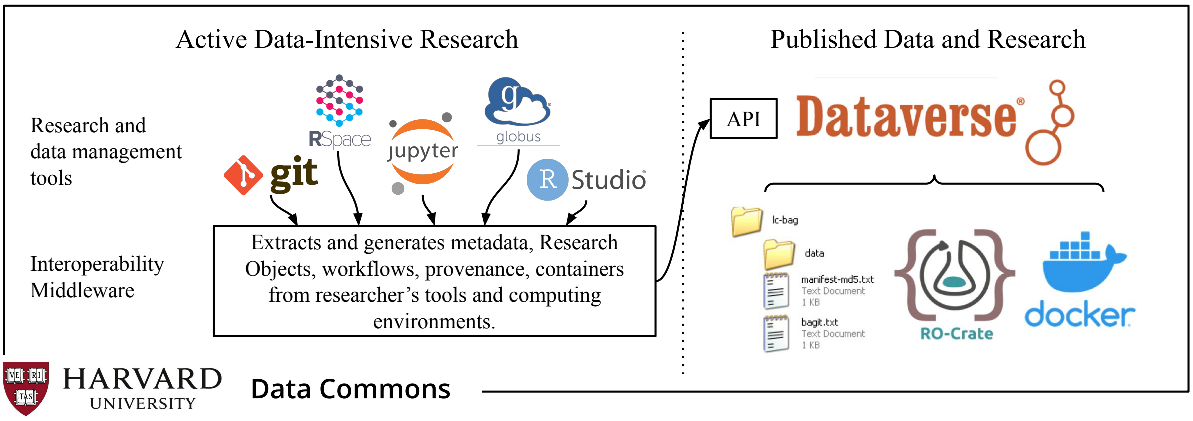

Active Data-Intensive Research
Published Data and Research
Research and data management tools
git
RSpace
jupyter
globus
R Studio
Interoperability Middleware
Extracts and generates metadata, Research Objects, workflows, provenance, containers from researcher's tools and computing environments.
API
Dataverse
lc-bag
data
manifest-md5.txt
Text Document
1 KB
bagit.txt
Text Document
1 KB
RO-Crate
docker
HARVARD UNIVERSITY
Data Commons

Figure 4.4: One aspect of Harvard Data Commons. Automatic encapsulation and deposit of artefacts from data management tools used during active research at the Harvard Dataverse repository.

stored in remote storage.

2. Support for advanced research workflows and providing packaging options for assets such as code and workflows in the Harvard Dataverse repository to enable reproducibility and reuse.

3. Integrating repositories supported by Harvard, which include DASH[84], the open access institutional repository, the Digital Repository Services (DRS) for preserving digital asset collections, and the Harvard Dataverse.

Particularly relevant to this article is the second objective of the Harvard Data Commons, which aims to support the deposit of research artefacts to Harvard Dataverse with sufficient information in the metadata to allow their future reuse (Figure 4.4). To support the incorporation of data, code, and other artefacts from various institutional infrastructures, Harvard Data Commons is currently working on RO-Crate adaptation. The RO-Crate metadata provides the necessary structure to make all research artefacts FAIR. The Dataverse software already has extensive support[85] for metadata, including the Data Documentation Initiative (DDI), Dublin Core, DataCite, and schema.org. Incorporating RO-Crate, which has the flexibility to describe a wide range of research resources, will facilitate their seamless transition from one infrastructure to the other within the Harvard Data Commons.

Even though the Harvard Data Commons is specific to Harvard University, the overall vision and the three objectives can be abstracted and applied to other universities or research organisations. The Commons will be designed and implemented using standards and commonly-used approaches to make it interoperable and reusable by others.

[84] https://dash.harvard.edu/
[85] https://guides.dataverse.org/en/latest/user/appendix.html

4.1.5 Related Work

With the increasing digitisation of research processes, there has been a significant call for the wider adoption of interoperable sharing of data and its associated metadata. We refer to [Koesten 2020] for a comprehensive overview and recommendations, in particular for data; notably that review highlights the wide variety of metadata and documentation that the literature prescribes for enabling data reuse. Likewise, we suggest [Leipzig 2021] that covers the importance of metadata standards in reproducible computational research.

Here we focus on approaches for bundling research artefacts along with their metadata. This notion of publishing compound objects for scholarly communication has a long history behind it [Claerbout 1992, Van de Sompel 2007], but recent approaches have followed three main strands: (i) publishing to centralised repositories; (ii) packaging approaches similar to RO-Crate; and (iii) bundling the computational workflow around a scientific experiment.

4.1.5.1 Bundling and Packaging Digital Research Artefacts

Early work making the case for publishing compound scholarly communication units [Van de Sompel 2007] led to the development of the Object Re-Use and Exchange model[86] (OAI-ORE), providing a structured **resource map** of the digital artefacts that together support a scholarly output.

The challenge of describing computational workflows was one of the main motivations for the early proposal of *Research Objects* (RO) [Bechhofer 2013] as first-class citizens for sharing and publishing. The RO approach involves bundling datasets, workflows, scripts and results along with traditional dissemination materials like journal articles and presentations, forming a single package. Crucially, these resources are not just gathered, but also individually typed, described and related to each other using semantic vocabularies. As pointed out in [Bechhofer 2013] an open-ended *Linked Data* approach is not sufficient for scholarly communication: a common data model is also needed in addition to common and best practices for managing and annotating lifecycle, ownership, versioning and attributions.

Considering the FAIR principles [Wilkinson 2016], we can say with hindsight that the initial RO approaches strongly targeted *Interoperability*, with a particular focus on the reproducibility of *in-silico experiments* involving computational workflows and the reuse of existing RDF vocabularies.

The first implementation of Research Objects for sharing workflows in myExperiment [Goble 2010] was based on RDF ontologies [Newman 2009], building on Dublin Core, FOAF, SIOC, Creative Commons and OAI-ORE to form myExperiment ontologies for describing social networking, attribution and credit, annotations, aggregation packs, experiments, view statistics, contributions, and workflow components [myExperiment 2009].

This initially workflow-centric approach was further formalised as the Wf4Ever Research Object Model [Belhajjame 2015], which is a general-purpose research artefact description framework.

[86] http://www.openarchives.org/ore/1.0/primer

This model is based on existing ontologies (FOAF, Dublin Core Terms, OAI-ORE and AO/OAC precursors to the W3C Web Annotation Model [Ciccarese 2017]) and adds specializations for workflow models and executions using W3C PROV-O [Lebo 2013a]. The Research Object statements are saved in a *manifest* (the OAI-ORE *resource map*), with additional annotation resources containing user-provided details such as title and description.

We now claim that one barrier for wider adoption of the Wf4Eer Research Object model for general packaging digital research artefacts was exactly this re-use of multiple existing vocabularies (FAIR principle I2: *Metadata use vocabularies that follow FAIR principles*), which in itself is recognised as a challenge [Katsumi 2016]. Adapters of the Wf4Ever RO model would have to navigate documentation of multiple overlapping ontologies, in addition to facing the usual Semantic Web development choices for RDF serialization formats, identifier minting and publishing resources on the Web.

Several developments for Research Objects improved on this situation, such as ROHub used by Earth Sciences [Garcia-Silva 2019], which provides a user-interface for making Research Objects, along with Research Object Bundle [Soiland-Reyes 2014] (RO Bundle), which is a ZIP-archive embedding data files and a JSON-LD serialization of the manifest with mappings for a limited set of terms. RO Bundle was also used for storing detailed workflow run provenance (TavernaPROV [Soiland-Reyes 2016]).

RO-Bundle evolved to Research Object BagIt archives[87], a variant of RO Bundle as a BagIt archive [Kunze 2018], used by Big Data Bags [Chard 2016], CWLProv [Khan 2019] and WholeTale [Chard 2020, Chard 2019].

4.1.5.2 FAIR Digital Objects

FAIR Digital Objects (FDO) [De Smedt 2020] have been proposed as a conceptual framework for making digital resources available in a Digital Objects (DO) architecture which encourages active use of the objects and their metadata. In particular, an FDO has five parts: (i) The FDO *content*, bit sequences stored in an accessible repository; (ii) a *Persistent Identifier* (PID) such as a DOI that identifies the FDO and can resolve these same parts; (iii) Associated rich *metadata*, as separate FDOs; (iv) Type definitions, also separate FDOs; (v) Associated *operations* for the given types. A Digital Object typed as a Collection aggregates other DOs by reference.

The Digital Object Interface Protocol [DONA 2018] can be considered an "abstract protocol" of requirements, DOs could be implemented in multiple ways. One suggested implementation is the FAIR Digital Object Framework,[88] based on HTTP and the Linked Data Principles. While there is agreement on using PIDs based on DOIs, consensus on how to represent common metadata, core types and collections as FDOs has not yet been reached. We argue that RO-Crate can play an important role for FDOs:

1. By providing a predictable and extensible serialisation of structured metadata.

[87] https://w3id.org/ro/bagit

[88] https://fairdigitalobjectframework.org/

2. By formalising how to aggregate digital objects as collections (and adding their context).
3. By providing a natural Metadata FDO in the form of the RO-Crate Metadata File.
4. By being based on Linked Data and schema.org vocabulary, meaning that PIDs already exist for common types and properties.

At the same time, it is clear that the goal of FDO is broader than that of RO-Crate; namely, FDOs are active objects with distributed operations, and add further constraints such as PIDs for every element. These features improve FAIR features of digital objects and are also useful for RO-Crate, but they also severely restrict the infrastructure that needs to be implemented and maintained in order for FDOs to remain accessible. RO-Crate, on the other hand, is more flexible: it can minimally be used within any file system structure, or ideally exposed through a range of Web-based scenarios. A *FAIR profile of RO-Crate* (e.g. enforcing PID usage) will fit well within a FAIR Digital Object ecosystem.

4.1.5.3 Packaging Workflows

The use of computational workflows, typically combining a chain of tools in an analytical pipeline, has gained prominence in particular in the life sciences. Workflows might be used primarily to improve computational scalability, as well as to assist in making computed data results FAIR [Goble 2020], for instance by improving reproducibility [Cohen-Boulakia 2017], but also because programmatic data usage help propagate their metadata and provenance [Kim 2008]. At the same time, workflows raise additional FAIR challenges, since they can be considered important research artefacts themselves. This viewpoint poses the problem of capturing and explaining the computational methods of a pipeline in sufficient machine-readable detail [Lamprecht 2019].

Even when researchers follow current best practices for workflow reproducibility [Grüning 2018b, Cohen-Boulakia 2017], the communication of computational outcomes through traditional academic publishing routes effectively adds barriers as authors are forced to rely on a textual manuscript representations. This hinder reproducibility and FAIR use of the knowledge previously captured in the workflow.

As a real-life example, let us look at a metagenomics article [Almeida 2019] that describes a computational pipeline. Here the authors have gone to extraordinary efforts to document the individual tools that have been reused, including their citations, versions, settings, parameters and combinations. The *Methods* section is two pages in tight double-columns with twenty four additional references, supported by the availability of data on an FTP server (60 GB) [EMBL-EBI 2019] and of open source code in GitHub Finn-Lab/MGS-gut[89] [EMBL-EBI 2020], including the pipeline as shell scripts and associated analysis scripts in R and Python.

This attention to reporting detail for computational workflows is unfortunately not yet the norm, and although bioinformatics journals have strong *data availability* requirements, they frequently

[89] https://github.com/Finn-Lab/MGS-gut

do not require authors to include or cite *software, scripts and pipelines* used for analysing and producing results [Soiland-Reyes 2020a]. Indeed, in the absence of a specific requirement and an editorial policy to back it up—such as eliminating the reference limit—authors are effectively discouraged from properly and comprehensively citing software [Nature 2019].

However detailed this additional information might be, another researcher who wants to reuse a particular computational method may first want to assess if the described tool or workflow is Re-runnable (executable at all), Repeatable (same results for original inputs on same platform), Reproducible (same results for original inputs with different platform or newer tools) and ultimately Reusable (similar results for different input data), Repurposable (reusing parts of the method for making a new method) or Replicable (rewriting the workflow following the method description) [Benureau 2017, Goble 2016].

Following the textual description alone, researchers would be forced to jump straight to evaluate "Replicable" by rewriting the pipeline from scratch. This can be expensive and error-prone. They would firstly need to install all the software dependencies and download reference datasets. This can be a daunting task, which may have to be repeated multiple times as workflows typically are developed at small scale on desktop computers, scaled up to local clusters, and potentially put into production using cloud instances, each of which will have different requirements for software installations.

In recent years the situation has been greatly improved by software packaging and container technologies like Docker and Conda, these technologies have been increasingly adopted in life sciences [Möller 2017] thanks to collaborative efforts such as BioConda [Grüning 2018a] and BioContainers [da Veiga Leprevost 2017], and support by Linux distributions (e.g. Debian Med [Möller 2010]). As of November 2021, more than 9,000 software packages are available in BioConda alone,[90] and 10,000 containers in BioContainers.[91]

Docker and Conda have been integrated into workflow systems such as Snakemake [Köster 2012], Galaxy [Afgan 2018] and Nextflow [Di Tommaso 2017], meaning a downloaded workflow definition can now be executed on a "blank" machine (except for the workflow engine) with the underlying analytical tools installed on demand. Even with using containers there is a reproducibility challenge, for instance Docker Hub's retention policy will expire container images after six months,[92] or a lack of recording versions of transitive dependencies of Conda packages could cause incompatibilities if the packages are subsequently updated.

These container and package systems only capture small amounts of metadata[93]. In particular, they do not capture any of the semantic relationships between their content. Understanding these relationships is made harder by the opaque wrapping of arbitrary tools with unclear

[90] https://anaconda.org/bioconda/

[91] https://biocontainers.pro/#/registry

[92] https://www.docker.com/blog/docker-hub-image-retention-policy-delayed-and-subscription-updates/

[93] Docker and Conda can use *build recipes*, a set of commands that construct the container image through downloading and installing its requirements. However these recipes are effectively another piece of software code, which may itself decay and become difficult to rerun.

functionality, licenses and attributions.

From this we see that computational workflows are themselves complex digital objects that need to be recorded not just as files, but in the context of their execution environment, dependencies and analytical purpose in research—as well as other metadata (e.g. version, license, attribution and identifiers).

It is important to note that having all these computational details in order to represent them in an RO-Crate is an ideal scenario—in practice there will always be gaps of knowledge, and exposing all provenance details automatically would require improvements to the data sources, workflow, workflow engine and its dependencies. RO-Crate can be seen as a flexible annotation mechanism for augmenting automatic workflow provenance. Additional metadata can be added manually, e.g. for sensitive clinical data that cannot be publicly exposed[94], or to cite software that lack persistent identifiers. This inline *FAIRifying* allows researchers to achieve "just enough FAIR" to explain their computational experiments.

4.1.6 Conclusion

RO-Crate has been established as an approach to packaging digital research artefacts with structured metadata. This approach assists developers and researchers to produce and consume FAIR archives of their research.

RO-Crate is formed by a set of best practice recommendations, developed by an open and broad community. These guidelines show how to use "just enough" standards in a consistent way. The use of structured metadata with a rich base vocabulary can cover general-purpose contextual relations, with a Linked Data foundation that ensures extensibility to domain- and application-specific uses. We can therefore consider an RO-Crate not just as a structured data archive, but as a multimodal scholarly knowledge graph that can help "FAIRify" and combine metadata of existing resources.

The adoption of simple Web technologies in the RO-Crate specification has helped a rapid development of a wide variety of supporting open source tools and libraries. RO-Crate fits into the larger landscape of open scholarly communication and FAIR Digital Object infrastructure, and can be integrated into data repository platforms. RO-Crate can be applied as a data/metadata exchange mechanism, assist in long-term archival preservation of metadata and data, or simply used at a small scale by individual researchers. Thanks to its strong community support, new and improved profiles and tools are being continuously added to the RO-Crate landscape, making it easier for adopters to find examples and support for their own use case.

4.1.6.1 Strictness vs flexibility

There is always a tradeoff between flexibility and strictness [Troncy 2010] when deciding on semantics of metadata models. Strict requirements make it easier for users and code to consume

[94] FAIR principle A2: *Metadata are accessible, even when the data are no longer available.* [Wilkinson 2016]

and populate a model, by reducing choices and having mandated "slots" to fill in. But such rigidity can also restrict richness and applicability of the model, as it in turn enforce the initial assumptions about what can be described.

RO-Crate attempts to strike a balance between these tensions, and provides a common metadata framework that encourages extensions. However, just like the RO-Crate specification can be thought of as a *core profile* of schema.org in JSON-LD, we cannot stress the importance of also establishing domain-specific RO-Crate profiles and conventions, as explored in Sections 4.1.2.4 on page 84 and 4.1.4 on page 93. Specialization comes hand-in-hand with the principle of *graceful degradation*; RO-Crate applications and users are free to choose the semantic detail level they participate at, as long as they follow the common syntactic requirements.

4.1.7 Future Work

The direction of future RO-Crate work is determined by the community around it as a collaborative effort. We currently plan on further outreach, building training material (including a comprehensive entry-level tutorial) and maturing the reference implementation libraries. We will also collect and build examples of RO-Crate *consumption*, e.g. Jupyter Notebooks that query multiple crates using knowledge graphs. In addition, we are exploring ways to support some entity types requested by users, e.g. detailed workflow runs or container provenance, which do not have a good match in schema.org. Such support could be added, for instance, by integrating other vocabularies or by having separated (but linked) metadata files.

Furthermore, we want to better understand how the community uses RO-Crate in practice and how it contrasts with other related efforts; this will help us to improve our specification and tools. By discovering commonalities in emerging usage (e.g. additional schema.org types), the community helps to reduce divergence that could otherwise occur with proliferation of further RO-Crate profiles. We plan to gather feedback via user studies, with the Linked Open Data community or as part of EOSC Bring-your-own-Data training events.

We operate in an open community where future and potential users of RO-Crate are actively welcomed to participate and contribute feedback and requirements. In addition, we are targeting a wider audience through extensive outreach activities[95] and by initiating new connections. Recent contacts include American Geophysical Union (AGU) on Data Citation Reliquary [Agarwal 2021], National Institute of Standards and Technology (NIST) on material science, and InvenioRDM[96] used by the Zenodo data repository. New Horizon Europe projects adapting RO-Crate include BY-COVID,[97] which aims to improve FAIR access to data on COVID-19 and other infectious diseases.

The main addition in the upcoming 1.2 release of the RO-Crate specifications will be the formalization of profiles[98] for different categories of crates. Additional entity types have been requested

[95] https://www.researchobject.org/ro-crate/outreach.html

[96] https://inveniosoftware.org/products/rdm/

[97] https://by-covid.org/

[98] https://www.researchobject.org/ro-crate/1.2-DRAFT/profiles

by users, e.g. workflow runs, business workflows, containers and software packages, tabular data structures; these are not always matched well with existing schema.org types, but may benefit from other vocabularies or even separate metadata files, e.g. from Frictionless Data.[99] We will be further aligning and collaborating with related research artefact description efforts like CodeMeta[100] for software metadata, Science-on-Schema.org[101] [Jones 2021] for datasets, FAIR Digital Objects[102] [De Smedt 2020] and activities in EOSC task forces[103] including the EOSC Interoperability Framework [Kurowski 2021].

[99] https://frictionlessdata.io/

[100] https://codemeta.github.io/

[101] https://science-on-schema.org/

[102] https://fairdo.org/

[103] https://www.eosc.eu/task-force-faq

4.2 Creating lightweight FAIR Digital Objects with RO-Crate

RO-Crate [Soiland-Reyes 2022a] (Section 4.1) is a lightweight method to package research outputs along with their metadata, based on Linked Data principles [Bizer 2009] and W3C standards. RO-Crate provides a flexible mechanism for researchers archiving and publishing rich data packages (or any other research outcome) by capturing their dependencies and context.

However, additional measures should be taken to ensure that a crate is also following the FAIR principles [Wilkinson 2016], including consistent use of persistent identifiers, provenance, community standards, clear machine/human-readable licensing for metadata and data, and Web publication of RO-Crates.

The FAIR Digital Object (FDO) approach [De Smedt 2020] gives a set of recommendations that aims to improve findability, accessibility, interoperability and reproducibility for any digital object, allowing implementation through different protocols or standards.

Here we present how we have followed the FDO recommendations and turned research outcomes into FDOs by publishing RO-Crates on the Web using HTTP, following best practices for Linked Data. We highlight challenges and advantages of the FDO approach, and reflect on what is required for an FDO profile to achieve FAIR RO-Crates.

The implementation allows for a broad range of use cases, across scientific domains. A minimal RO-Crate may be represented as a persistent URI resolving to a summary website describing the outputs in a scientific investigation (e.g. https://w3id.org/dgarijo/ro/sepln2022 with links to the used datasets along with software).

One of the advantages of RO-Crates is flexibility, particularly regarding the metadata accompanying the actual research outcome. RO-Crate extends [schema.org], a popular vocabulary for describing resources on the Web [Guha 2016]. A generic RO-Crate is not required to be typed beyond `Dataset`[104] In practice, RO-Crates declare conformance to particular profiles[105], allowing processing based on the specific needs and assumptions of a community or usage scenario. This, effectively, makes RO-Crates typed and thus machine-actionable. RO-Crate profiles serve as metadata templates, making it easier for communities to agree and build upon their own metadata needs.

RO-Crates have been combined with *machine-actionable Data Management Plans* (maDMPs) to automate and facilitate management of research data [Miksa 2020]. This mapping allows RO-Crates to be generated out of maDMPs and vice versa. The ELIXIR Software Management Plans [Alves 2021] is planning to move their questionnaire to a machine-actionable format with RO-Crate. ELIXIR Biohackathon[106] 2022 will[107] explore[108] integration of RO-Crate and the Data

[104] Resources described by an RO-Crate are also typed, e.g. `Person`, `Organization`, `ScholarlyArticle`, `ImageObject`. https://www.researchobject.org/ro-crate/1.1/contextual-entities.html

[105] https://www.researchobject.org/ro-crate/profiles.html

[106] https://biohackathon-europe.org/

[107] See report [Eguinoa 2023]

[108] https://github.com/elixir-europe/biohackathon-projects-2022/tree/main/10

Stewardship Wizard[109] [Pergl 2019] with Galaxy, which can automate FDO creation that also follows data management plans.

A tailored RO-Crate profile has been defined to represent Electronic Lab Notebooks (ELN) protocols bundled together with metadata and related datasets. [Schröder 2022] uses RO-Crates to encode provenance information at different levels, including researchers, manufacturers, biological and chemical resources, activities, measurements, and resulting research data. The use of RO-Crates makes it easier to programmatically question-answer information related to the protocols, for instance activities, resources and equipment used to create data.

Another example is WorkflowHub[110] [Goble 2021] which defines the Workflow RO-Crate[111] profile [Bacall 2022], imposing additional constraints such as the presence of a main workflow and a license. It also specifies which entity types and properties must be used to provide such information, implicitly defining a set of operations (e.g., get the main workflow and its language) that are valid on all complying crates. The workflow system Galaxy [Galaxy 2022] retrieves such Workflow Crates using GA4GH TRS API[112].

The workflow profile has been further extended (with OOP-like inheritance) in Workflow Testing[113] RO-Crate, adding formal workflow testing components: this adds operations such as getting remote test instances and test definitions, used by the LifeMonitor[114] service to keep track of the health status of multiple published workflows.

While RO-Crates use Web technologies, they are also *self-contained*, moving data along with their metadata. This is a powerful construct for interoperability across FAIR repositories, but this raises some challenges with regards to mutability and persistence of crates.

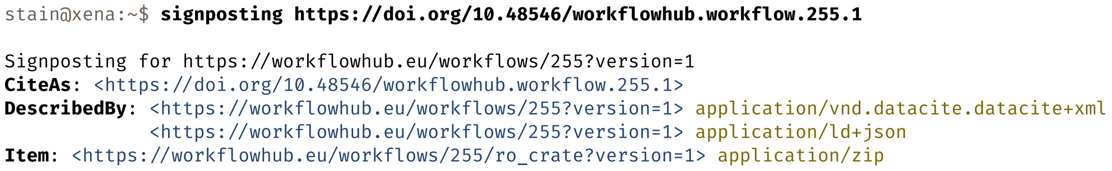
To illustrate how such challenges can be handled, we detail how the WorkflowHub repository follows several FDO principles[115]:

1. Workflow entries must be *frozen* for editing and have complete kernel metadata (title, authors, license, description) [FDOF4] before they can be assigned a persistent identifier, e.g. https://doi.org/10.48546/workflowhub.workflow.255.1 [FDOF1]

2. Computational workflows can be composed of multiple files used as a whole, e.g. CWL files in a GitHub repository. These are snapshotted as a single RO-Crate ZIP, indicating the main workflow. [FDOF11]

3. PID resolution can content-negotiate to Datacite's PID metadata [FDOF2] or use FAIR Signposting[116] to find an RO-Crate containing the workflow [FDOF3] and richer JSON-LD metadata resources [FDOF5,FDOF8], see Figure 4.5 on the facing page.

[109] https://ds-wizard.org/
[110] https://workflowhub.eu/
[111] https://w3id.org/workflowhub/workflow-ro-crate/1.0
[112] https://about.workflowhub.eu/developer/trs/
[113] https://crs4.github.io/life_monitor/workflow_testing_ro_crate
[114] https://www.lifemonitor.eu/
[115] See Table 2.1.1 on page 17
[116] https://signposting.org/FAIR/



```
stain@xena:~$ signposting https://doi.org/10.48546/workflowhub.workflow.255.1

Signposting for https://workflowhub.eu/workflows/255?version=1
CiteAs: <https://doi.org/10.48546/workflowhub.workflow.255.1>
DescribedBy: <https://workflowhub.eu/workflows/255?version=1> application/vnd.datacite.datacite+xml
             <https://workflowhub.eu/workflows/255?version=1> application/ld+json
Item: <https://workflowhub.eu/workflows/255/ro_crate?version=1> application/zip
```

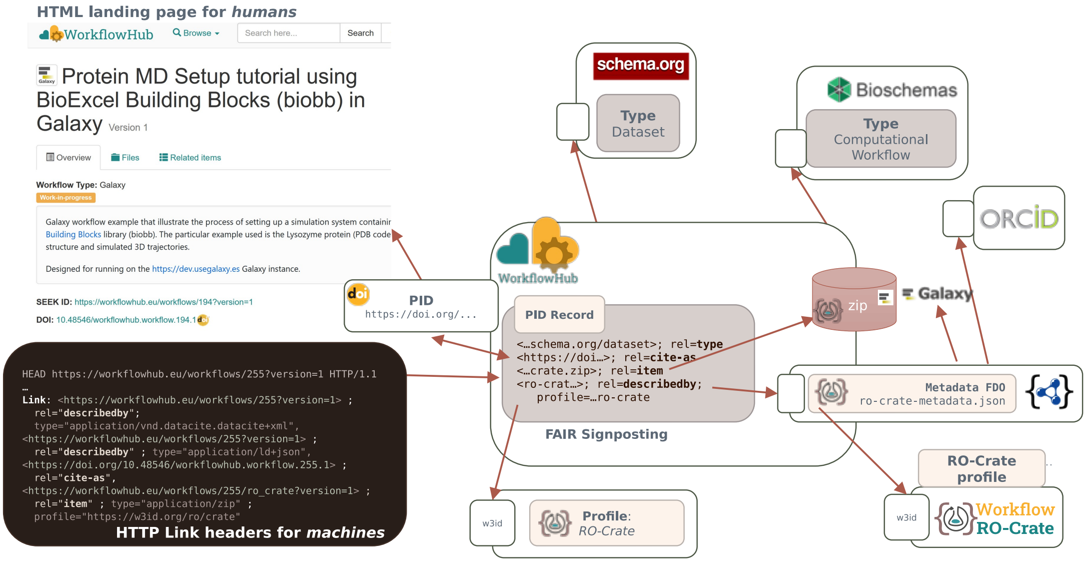
Figure 4.5: FAIR Signposting. FAIR Signposting on a workflow PID [Bayarri 2022] discovered from HTTP `Link:` headers using the Signposting tool[120] shows machine-actionable navigation to content-negotiate for the metadata FDOs, as well as download bit sequence (FDOF3) as an RO-Crate ZIP. JSON-LD[121] from workflowhub.eu follows the BioSchemas ComputationalWorkflow profile[122] to give workflow details not included in DataCite's general JSON-LD[123].

4. Metadata uses schema.org [FDOF7] following the community-developed Bioschemas ComputationalWorkflow[117] profile [FDOF10].
5. Workflows are discovered using theGA4GH TRS API[118] [FDOF5,FDOF6,FDOF11] and created/modified using CRUD operations[119] [FDOF6]
6. The RO-Crate profile, effectively the FDO Type [FDOF7], is declared as https://w3id.org/workflowhub/workflow-ro-crate/1.0; the workflow language (e.g. https://w3id.org/workflowhub/workflow-ro-crate#galaxy is defined in metadata of the main workflow.

Further work on RO-Crate profiles include to formalise links to the API operations and repositories (FDOF5,FDOF7), to include PIDs of profiles and types in the FAIR Signposting, and HTTP navigation to individual resources within the RO-Crate.

RO-Crate has shown a broad adoption by communities across many scientific disciplines, providing a lightweight, and therefore easy to adopt, approach to generating FAIR Digital Objects (Figure 4.6 on the next page). It is rapidly becoming an integral part of the interoperability fabric between the different components as demonstrated here for WorkflowHub, contributing to building the European Open Science Cloud.

[117] https://bioschemas.org/profiles/ComputationalWorkflow/1.0-RELEASE
[118] https://about.workflowhub.eu/developer/trs/
[119] https://workflowhub.eu/api
[120] https://pypi.org/project/signposting/
[121] https://workflowhub.eu/workflows/255.jsonld
[122] https://bioschemas.org/profiles/ComputationalWorkflow/1.0-RELEASE
[123] https://data.crosscite.org/application/ld+json/10.48546/workflowhub.workflow.255.1



HTML landing page for *humans*

Protein MD Setup tutorial using BioExcel Building Blocks (biobb) in Galaxy Version 1

Workflow Type: Galaxy

Work-in-progress

Galaxy workflow example that illustrate the process of setting up a simulation system containir Building Blocks library (biobb). The particular example used is the Lysozyme protein (PDB code structure and simulated 3D trajectories.

Designed for running on the https://dev.usegalaxy.es Galaxy instance.

SEEK ID: https://workflowhub.eu/workflows/194?version=1

DOI: 10.48546/workflowhub.workflow.194.1

```
HEAD https://workflowhub.eu/workflows/255?version=1 HTTP/1.1
…
Link: <https://workflowhub.eu/workflows/255?version=1> ;
  rel="describedby";
  type="application/vnd.datacite.datacite+xml",
<https://workflowhub.eu/workflows/255?version=1> ;
  rel="describedby" ; type="application/ld+json",
<https://doi.org/10.48546/workflowhub.workflow.255.1> ;
  rel="cite-as",
<https://workflowhub.eu/workflows/255/ro_crate?version=1> ;
  rel="item" ; type="application/zip" ;
  profile="https://w3id.org/ro/crate"
```

HTTP Link headers for *machines*

PID Record

```
<…schema.org/dataset>; rel=type
<https://doi…>; rel=cite-as
<…crate.zip>; rel=item
<ro-crat…>; rel=describedby;
   profile=…ro-crate
```

Figure 4.6: WorkflowHub FDOs using Signposting and RO-Crate. In this implementation of FDO (compare with Figure 2.1 on page 17), WorkflowHub uses DOIs for persistent identifiers, which for human readers resolve to a landing page. Machine clients can extract the FAIR Signposting `Link` headers [Van de Sompel 2022] to form the FDO PID Record. Link relation `cite-as` provides the PID [Bayarri 2022] (in case the WorkflowHub page was discovered in other ways), and `describedby` links to the metadata FDO in JSON-LD format. Within the metadata file it indicates conformance to the Workflow RO-Crate profile (a Profile Crate FDO), while `item` links to the downloadable ZIP archive, which contains both the Galaxy workflow files and the RO-Crate metadata file. Alternative metadata in XML following the DataCite Metadata Schema is also linked to using `describedby`. Link relation `type` in the Signposting can provide the FDO type; this is not yet implemented by WorkflowHub—it is currently unclear if this type should be a *Dataset* (the download from this landing page is an RO-Crate) and/or *ComputationalWorkflow* (the PID/page/crate identifies and describes a workflow).

4.3 Formalizing RO-Crate in First Order Logic

Below is a formalization of the concept of RO-Crate as a set of relations using First Order Logic:

4.3.1 Language

Definition of language $\mathcal{L}_{rocrate}$:

$$\begin{aligned}
\mathcal{L}_{rocrate} &= \{Property(p), Class(c), Value(x), \mathbb{R}, \mathbb{S}\} \\
\mathbb{D} &= \mathbb{IRI} \\
\mathbb{IRI} &\equiv \text{IRIs as defined in RFC3987 [Dürst 2005]} \\
\mathbb{R} &\equiv \text{real or integer numbers} \\
\mathbb{S} &\equiv \text{literal strings}
\end{aligned}$$

The domain of discourse $\mathbb{D}$ is the set of $\mathbb{IRI}$ identifiers (notation `<http://example.com/>`)[124], with additional descriptions using numbers $\mathbb{R}$ (notation 13.37) and literal strings $\mathbb{S}$ (notation "Hello").

From this formalised language $\mathcal{L}_{rocrate}$ we can interpret an RO-Crate in any representation that can gather these descriptions, their properties, classes, and literal attributes.

4.3.2 Minimal RO-Crate

The definitions on the following page use $\mathcal{L}_{rocrate}$ for a minimal[125] RO-Crate:

[124] For simplicity, blank nodes are not included in this formalisation, as RO-Crate recommends the use of IRI identifiers: https://www.researchobject.org/ro-crate/1.1/appendix/jsonld.html#describing-entities-in-json-ld

[125] The full list of types, relations and attribute properties from the RO-Crate specification are not included. Examples shown include $datePublished$, $CreativeWork$ and $name$.

$$
\begin{aligned}
ROCrate(R) \quad &\vDash \quad Root(R) \wedge Mentions(R,R) \wedge hasPart(R,d) \wedge \\
&\qquad Mentions(R,d) \wedge DataEntity(d) \wedge \\
&\qquad Mentions(R,c) \wedge ContextualEntity(c) \\
\forall r \, Root(r) \quad &\Rightarrow \quad Dataset(r) \wedge name(r,n) \wedge description(r,d) \wedge \\
&\qquad datePublished(r,date) \wedge license(e,l) \\
\forall e \forall n \, name(e,n) \quad &\Rightarrow \quad Value(n) \\
\forall e \forall s \, description(e,s) \quad &\Rightarrow \quad Value(s) \\
\forall e \forall d \, datePublished(e,d) \quad &\Rightarrow \quad Value(d) \\
\forall e \forall l \, license(e,l) \quad &\Rightarrow \quad ContextualEntity(l) \\
DataEntity(e) \quad &\equiv \quad File(e) \oplus Dataset(e) \\
Entity(e) \quad &\equiv \quad DataEntity(e) \vee ContextualEntity(e) \\
\forall e \, Entity(e) \quad &\Rightarrow \quad type(e,c) \wedge Class(c) \\
\forall e \, ContextualEntity(e) \quad &\Rightarrow \quad name(e,n) \\
Mentions(R,s) \quad &\vDash \quad Relation(s,p,e) \oplus Attribute(s,p,l) \\
Relation(s,p,o) \quad &\vDash \quad Entity(s) \wedge Property(p) \wedge Entity(o) \\
Attribute(s,p,v) \quad &\vDash \quad Entity(s) \wedge Property(p) \wedge Value(v) \\
Value(v) \quad &\equiv \quad v \in \mathbb{R} \oplus v \in \mathbb{S}
\end{aligned}
$$

An $ROCrate(R)$ is defined as a self-described *Root Data Entity*, which contains parts (*data entities*), which are further described in *contextual entities*. These terms align with their use in the RO-Crate 1.1 terminology[126].

The $Root(r)$ is a type of $Dataset(r)$, and must as metadata have at least the attributes $name$, $description$ and $datePublished$, as well as a contextual entity that identify its $license$. These predicates correspond to the RO-Crate 1.1 minimal requirements for the root data entity[127].

The concept of an $Entity(e)$ is introduced as being either a $DataEntity(e)$, a $ContextualEntity(e)$, or both[128]. Any $Entity(e)$ must be typed with at least one $Class(c)$, and every $ContextualEntity(e)$ must also have a $name(e,n)$; this corresponds to expectations for any *referenced contextual entity* (Section 4.1.2.2 on page 82).

For simplicity in this formalization (and to assist production rules below) R is a constant representing a single RO-Crate, typically written to independent RO-Crate Metadata files. R is used by $Mentions(R,e)$ to indicate that e is an Entity described by the RO-Crate and therefore its metadata (a set of $Relation$ and $Attribute$ predicates) form part of the RO-Crate serialization. $Relation(s,p,o)$ and $Attribute(s,p,x)$ are defined as a *subject–predicate–object* triple pattern from an $Entity(s)$ using a $Property(p)$ to either another $Entity(o)$ or a $Value(x)$ value.

[126] https://www.researchobject.org/ro-crate/1.1/terminology

[127] https://www.researchobject.org/ro-crate/1.1/root-data-entity.html#direct-properties-of-the-root-data-entity

[128] https://www.researchobject.org/ro-crate/1.1/contextual-entities.html#contextual-vs-data-entities

4.3.3 Example of formalised RO-Crate

The below is an example RO-Crate represented using the above formalisation, assuming a base IRI of `<http://example.com/ro/123/>`:

$ROCrate(\texttt{<http://example.com/ro/123/>})$

$name(\texttt{<http://example.com/ro/123/>},$
"Data files associated with the manuscript:Effects of ...")

$description(\texttt{<http://example.com/ro/123/},$
"Palliative care planning for nursing home residents ...")

$license(\texttt{<http://example.com/ro/123/>},$
$\texttt{<https://spdx.org/licenses/CC-BY-4.0>})$

$datePublished(\texttt{<http://example.com/ro/123/>},$ "2017-02-23")

$hasPart(\texttt{<http://example.com/ro/123/>},$
$\texttt{<http://example.com/ro/123/file.txt>})$

$hasPart(\texttt{<http://example.com/ro/123/>},$
$\texttt{<http://example.com/ro/123/interviews/>})$

$ContextualEntity(\texttt{<https://spdx.org/licenses/CC-BY-4.0>})$

$name(\texttt{<https://spdx.org/licenses/CC-BY-4.0>},$
"Creative Commons Attribution 4.0")

$ContextualEntity(\texttt{<https://spdx.org/licenses/CC-BY-NC-4.0>})$

$name(\texttt{<https://spdx.org/licenses/CC-BY-NC-4.0>},$
"Creative Commons Attribution Non Commercial 4.0")

$File(\texttt{<http://example.com/ro/123/survey.csv>})$

$name(\texttt{<http://example.com/ro/123/survey.csv>},$ "Survey of care providers")

$Dataset(\texttt{<http://example.com/ro/123/interviews/>})$

$name(\texttt{<http://example.com/ro/123/interviews/>},$
"Audio recordings of care provider interviews")

$license(\texttt{<http://example.com/ro/123/interviews/>},$
$\texttt{<https://spdx.org/licenses/CC-BY-NC-4.0>})$

Notable from this triple-like formalization is that a RO-Crate R is fully represented as a tree at depth 2 helped by the use of $\mathbb{IRI}$ nodes. For instance the aggregation from the root entity $hasPart($...`interviews/>`$)$ is at same level as the data entity's property

$license($…`CC-BY-NC-4.0>`$)$ and that contextual entity's attribute $name($...Non Commercial 4.0”$)$. As shown in Section 4.1.2.5 on page 86, the RO-Crate Metadata File serialization is an equivalent shallow tree, although at depth 3 to cater for the JSON-LD preamble of `"@context"` and `"@graph"`.

In reality many additional attributes and contextual types from Schema.org types like http://schema.org/affiliation and http://schema.org/Organization would be used to further describe the RO-Crate and its entities, but as these are optional (*SHOULD* requirements) they do not form part of this formalization.

4.3.4 Mapping to RDF with Schema.org

A formalised RO-Crate in $\mathcal{L}_{rocrate}$ can be mapped to different serializations. Assume a simplified[129] language $\mathcal{L}_{RDF}$ based on the RDF abstract syntax [RDF 1.1 2014]:

$$\begin{aligned}
\mathcal{L}_{RDF} &\equiv \{Triple(s,p,o), IRI(i), BlankNode(b), Literal(s), \mathbb{IRI}, \mathbb{S}, \mathbb{R}\}\\
\mathbb{D}_{RDF} &\equiv \mathbb{S}\\
\forall i\, IRI(i) &\Rightarrow i \in \mathbb{IRI}\\
\forall s \forall p \forall o\, Triple(s,p,o) &\Rightarrow \Big(IRI(s) \lor BlankNode(s)\Big) \land\\
& \quad\; IRI(p) \land\\
& \quad\; \Big(IRI(o) \lor BlankNode(o) \lor Literal(o)\Big)\\
Literal(v) &\models Value(v) \land Datatype(v,t) \land IRI(t)\\
\forall v\, Value(v) &\Rightarrow v \in \mathbb{S}\\
LanguageTag(v,l) &\equiv Datatype(v,\\
& \quad\; \texttt{<http://www.w3.org/1999/02/22-rdf-syntax-ns\#langString>})
\end{aligned}$$

Below follows a mapping from $\mathcal{L}_{rocrate}$ to $\mathcal{L}_{RDF}$ using Schema.org as vocabulary:

$$\begin{aligned}
Property(p) &\Rightarrow type(p, \texttt{<http://www.w3.org/2000/01/rdf-schema\#Property>})\\
Class(c) &\Rightarrow type(c, \texttt{<http://www.w3.org/2000/01/rdf-schema\#Class>})\\
Dataset(d) &\Rightarrow type(d, \texttt{<http://schema.org/Dataset>})\\
File(f) &\Rightarrow type(f, \texttt{<http://schema.org/MediaObject>})\\
ContextualEntity(e) &\Rightarrow type(f, \texttt{<http://schema.org/Thing>})\\
CreativeWork(e) &\Rightarrow ContextualEntity(e)\\
& \quad\; \land type(e, \texttt{<http://schema.org/CreativeWork>})\\
hasPart(e,t) &\Rightarrow Relation(e, \texttt{<http://schema.org/hasPart>}, t)
\end{aligned}$$

[129] This simplification and mapping does not cover the extensive list of literal datatypes built into RDF 1.1, only strings and decimal real numbers. Likewise, $LanguageTag$ is deliberately not utillised below.

$$
\begin{aligned}
name(e,n) &\Rightarrow Attribute(e, \texttt{<http://schema.org/name>}, n)\\
description(e,s) &\Rightarrow Attribute(e, \texttt{<http://schema.org/description>}, s)\\
datePublished(e,d) &\Rightarrow Attribute(e, \texttt{<http://schema.org/datePublished>}, d)\\
license(e,l) &\Rightarrow Relation(e, \texttt{<http://schema.org/license>}, l) \land CreativeWork(l)\\
type(e,t) &\Rightarrow Relation(e, \texttt{<http://www.w3.org/1999/02/22-rdf-syntax-ns\#type>}, t)\\
&\quad \land Class(t)\\
String(s) &\equiv Value(s) \land s \in \mathbb{S}\\
String(s) &\Rightarrow Datatype(s, \texttt{<http://www.w3.org/2001/XMLSchema\#string>})\\
Decimal(d) &\equiv Value(d) \land d \in \mathbb{R}\\
Decimal(d) &\Rightarrow Datatype(d, \texttt{<http://www.w3.org/2001/XMLSchema\#decimal>})\\
Relation(s,p,o) &\Rightarrow Triple(s,p,o) \land IRI(s) \land IRI(o)\\
Attribute(s,p,o) &\Rightarrow Triple(s,p,o) \land IRI(s) \land Literal(o)
\end{aligned}
$$

Note that in the JSON-LD serialization of RO-Crate, the expression of $Class$ and $Property$ is typically indirect: The JSON-LD `@context` maps to Schema.org IRIs, which, when resolved as Linked Data, embed their formal definition as RDFa. Extensions may, however, include such term definitions directly in the RO-Crate.

4.3.5 RO-Crate 1.1 Metadata File Descriptor

An important RO-Crate principle is that of being **self-described**. Therefore, the serialisation of the RO-Crate into a file should also describe itself in a Metadata File Descriptor[130], indicating it is *about* (describing) the RO-Crate root data entity, and that it $conformsTo$ a particular version of the RO-Crate specification:

$$
\begin{aligned}
about(s,o) &\Rightarrow Relation(s, \texttt{<http://schema.org/about>}, o)\\
conformsTo(s,o) &\Rightarrow Relation(s, \texttt{<http://purl.org/dc/terms/conformsTo>}, o)\\
MetadataFile(m) &\Rightarrow CreativeWork(m) \land about(m,R) \land ROCrate(R) \land\\
&\quad conformsTo(m, \texttt{<https://w3id.org/ro/crate/1.1>})
\end{aligned}
$$

Note that although the metadata file necessarily is an *information resource* written to disk or served over the network (as JSON-LD), it is not considered to be a contained *part* of the RO-Crate in the form of a *data entity*, rather it is described only as a *contextual entity*.

In the conceptual model, the *RO-Crate Metadata File* can be seen as the top-level node that describes the *RO-Crate Root*; however, in the formal model (and the JSON-LD format) the metadata file descriptor is an additional contextual entity that is not affecting the depth-limit of the RO-Crate.

[130] https://www.researchobject.org/ro-crate/1.1/root-data-entity.html#ro-crate-metadata-file-descriptor

4.3.6 Forward-chained Production Rules for JSON-LD

Combining the above predicates and Schema.org mapping with rudimentary JSON templates, these forward-chaining production rules can output JSON-LD according to the RO-Crate 1.1 specification[131]:

$$
\begin{array}{rcl}
Mentions(R,s) \land Relation(s,p,o) & \Rightarrow & Mentions(R,o) \\
IRI(i) & \Rightarrow & \texttt{"}i\texttt{"} \\
Decimal(d) & \Rightarrow & d \\
String(s) & \Rightarrow & \texttt{"}s\texttt{"} \\
\forall e \forall t \; type(e,t) & \Rightarrow & \texttt{\{"@id":}e\texttt{,} \\
 & & \;\texttt{"@type":}t \\
 & & \texttt{\}} \\
\forall s \forall p \forall o \; Relation(s,p,o) & \Rightarrow & \texttt{\{"@id":}s\texttt{,} \\
 & & \; p\texttt{: \{ "@id":}o\texttt{\}} \\
 & & \texttt{\}} \\
\forall s \forall p \forall v \; Attribute(s,p,v) & \Rightarrow & \texttt{\{"@id":}s\texttt{,} \\
 & & \; p\texttt{:}v \\
 & & \texttt{\}} \\
\forall r \forall c ROCrate(r) & \Rightarrow & \texttt{\{ "@graph": [} \\
 & & \quad Mentions(r,c) * \\
 & & \;\; \texttt{]} \\
 & & \texttt{\}} \\
R & \equiv & \texttt{<./>} \\
R & \Rightarrow & MetadataFile(\texttt{<ro-crate-metadata.json>})
\end{array}
$$

This exposes the first order logic domain of discourse of IRIs, with rational numbers and strings as their corresponding JSON-LD representation. These production rules first grow the graph of R by adding a transitive rule—anything described in R which is related to o, means that o is also mentioned by the $ROCrate(R)$. For simplicity this rule is one-way; in theory the graph can also contain free-standing contextual entities that have outgoing relations to data- and contextual entities, but these are proposed to be bound to the root data entity with Schema.org relation mentions[132].

[131] **Limitations:** Contextual entities not related from the RO-Crate (e.g. using inverse relations to a data entity) would not be covered by the single direction $Mentions(R, s)$ production rule; see GitHub issue ResearchObject/ro-crate#122 (https://github.com/ResearchObject/ro-crate/issues/122).
The $datePublished(e, d)$ rule do not include syntax checks for the ISO 8601 datetime format. Compared with RO-Crate examples, this generated JSON-LD does not use a $@context$ as the IRIs are produced unshortened, a post-step could do JSON-LD Flattening with a versioned RO-Crate context. The `@type` expansion is included for clarity, even though this is also implied by the $type(e, t)$ expansion to $Relation(e, \texttt{xsd:type})$.

[132] http://schema.org/mentions

5 Computational Workflows

In order to investigate **RQ3** (on page 11), and considering important parts of the FAIR principles are *Reuse* and *provenance*, this chapter examines in closer details how FAIR Digital Objects and RO-Crate can be used with **Computational Workflows**.

Section 5.1 on the facing page proposes that tools in computational workflows, when wrapped as interoperable building blocks, can be considered as FAIR Digital Objects, with a use case from biomolecular simulation.

Sections 5.2 on page 133 and 5.3 on page 148 explore how FDOs and Research Objects can be constructed incrementally using computational workflows, with a use case from specimen digitization in natural history collections.

Section 5.4 on page 152 presents a profile of RO-Crate to capture workflow execution provenance, with incremental granularity levels and six workflow engine implementations. Use cases include machine learning-aided tumour detection and compatibility with PROV approaches.

Supplementary materials that may assist readers of this chapter provide further details on FAIR Computational Workflows [Goble 2020], WorkflowHub[1] [Goble 2021], Common Workflow Language[2] [Crusoe 2022] and making a software tool workflow ready[3] [Brack 2022a].

On the aspects of workflow provenance, recommended reading in supplementary materials covers CWLProv [Khan 2019], RO-Crate in Galaxy[4] [De Geest 2022] and Common Provenance Model [Wittner 2020, Wittner 2023a].

[1] https://s11.no/2021/phd/workflow-collaboratory/

[2] https://s11.no/2022/phd/methods-included/

[3] https://s11.no/2022/phd/10-simple-rules-for-workflow-tools/

[4] https://s11.no/2022/phd/galaxy-ro-crate/

5.1 Making Canonical Workflow Building Blocks interoperable across workflow languages

We introduce the concept of *Canonical Workflow Building Blocks* (CWBB), a methodology of describing and wrapping computational tools, in order for them to be utilised in a reproducible manner from multiple workflow languages and execution platforms. The concept is implemented and demonstrated with the BioExcel Building Blocks library (BioBB), a collection of tool wrappers in the field of computational biomolecular simulation. Interoperability across different workflow languages is showcased through a protein Molecular Dynamics setup transversal workflow, built using this library and run with 5 different Workflow Management Systems (WfMSs) We argue such practice is a necessary requirement for FAIR Computational Workflows and an element of Canonical Workflow Frameworks for Research (CWFR) in order to improve widespread adoption and reuse of computational methods across workflow language barriers.

5.1.1 Introduction

The need for *reproducibility* of research software usage is well established [Stodden 2016, Leipzig 2021, Katz 2021a], and adaptation of *workflow management systems* (**WfMS**) together with *software packaging and containers* [Möller 2017] have been proposed as key ingredients for making research software usage FAIR and reproducible [Cohen-Boulakia 2017, Grüning 2018b, Lamprecht 2019]. Recently it is also argued that computational workflows should also be treated as FAIR Digital Objects (FDOs) [De Smedt 2020] in their own right, with identifier, metadata [Leipzig 2021] and interoperability requirements [Goble 2020].

BioExcel[5], a European Centre of Excellence for Computational Biomolecular Research, has a particular focus on the research domains molecular dynamics simulations and bioinformatics with use of *High Performance Computing* (**HPC**) to approach Exascale performance, while also improving usability. The *BioExcel Building Blocks* (**BioBB**) [Andrio 2019] have been created as portable wrappers of open-source computational tools identified as useful for BioExcel workflows, forming several *families* of documented and interoperable operations that can be called from multiple workflow systems. This interoperability is shown with the BioBB demonstrator workflows, along with multiple tutorials and notebooks.

We propose that these building blocks and their families can themselves be considered *composite Digital Objects*: collections of software packages and their source code, guides and tutorials, as well as workflow management system integrations and workflow examples. In addition, the building blocks, as wrappers of upstream open source tools, benefit from and refer to the tools' existing documentation, support forums, academic publications and wider development context.

Given BioBB as a starting point, we define a generalised methodology of *Canonical Workflow Building Blocks* (**CWBB**), through the definition of a set of requirements and recommendations

[5] https://bioexcel.eu/

for how to formalise and develop a family of compatible computational tools as Digital Objects. These building blocks let researchers instantiate a Canonical Workflow in multiple workflow management systems, while also benefiting from the FAIR aspects of the CWBB Digital Objects.

5.1.2 Methods

The BioExcel Building Blocks[6] library [Andrio 2019], created and implemented within the BioExcel CoE, is a collection of portable wrappers of common biomolecular simulation tools. The BioBB library is designed to i) increase the *interoperability* between the tools wrapped; ii) *ease the implementation* of biomolecular simulation workflows; and iii) increase the *reusability and reproducibility* of the generated workflows. To achieve these main goals, the library was designed following the FAIR principles for research software development best practices [Lamprecht 2019].

The result is a collection of building block modules, divided in sets of tool wrappers focused on similar functionalities (e.g. Molecular Dynamics, Virtual Screening). Each of the modules is built from a combination of:

(i) software packaging (Pip[7], BioConda[8], BioContainers[9])

(ii) documentation (ReadTheDocs[10])

(iii) interactive tutorials (Jupyter Notebooks[11], Binder[12])

(iv) registry & findability (bio.tools[13], BioSchemas[14], WorkflowHub[15])

(v) WfMS integration stubs (CWL[16], Galaxy[17], PyCOMPSs[18])

(vi) source Code (GitHub[19])

(vii) REST APIs (OpenAPI[20], Swagger[21])

Notably, all building blocks follow the same pattern of installation, configuration and interaction.

[6] http://mmb.irbbarcelona.org/biobb/

[7] https://pypi.org/project/biobb/

[8] https://bioconda.github.io/search.html?q=biobb

[9] https://biocontainers.pro/

[10] https://biobb.readthedocs.io/

[11] http://mmb.irbbarcelona.org/biobb/workflows/tutorials/md_setup

[12] https://bioexcel-binder.tsi.ebi.ac.uk/v2/gh/bioexcel/biobb_wf_md_setup/master?filepath=biobb_wf_md_setup%2Fnotebooks%2Fbiobb_MDsetup_tutorial.ipynb

[13] https://bio.tools/biobb

[14] https://bioschemas.org/profiles/ComputationalTool/0.5-DRAFT/

[15] https://workflowhub.eu/programmes/2

[16] https://github.com/bioexcel/biobb_adapters/tree/v0.1.4/biobb_adapters/cwl

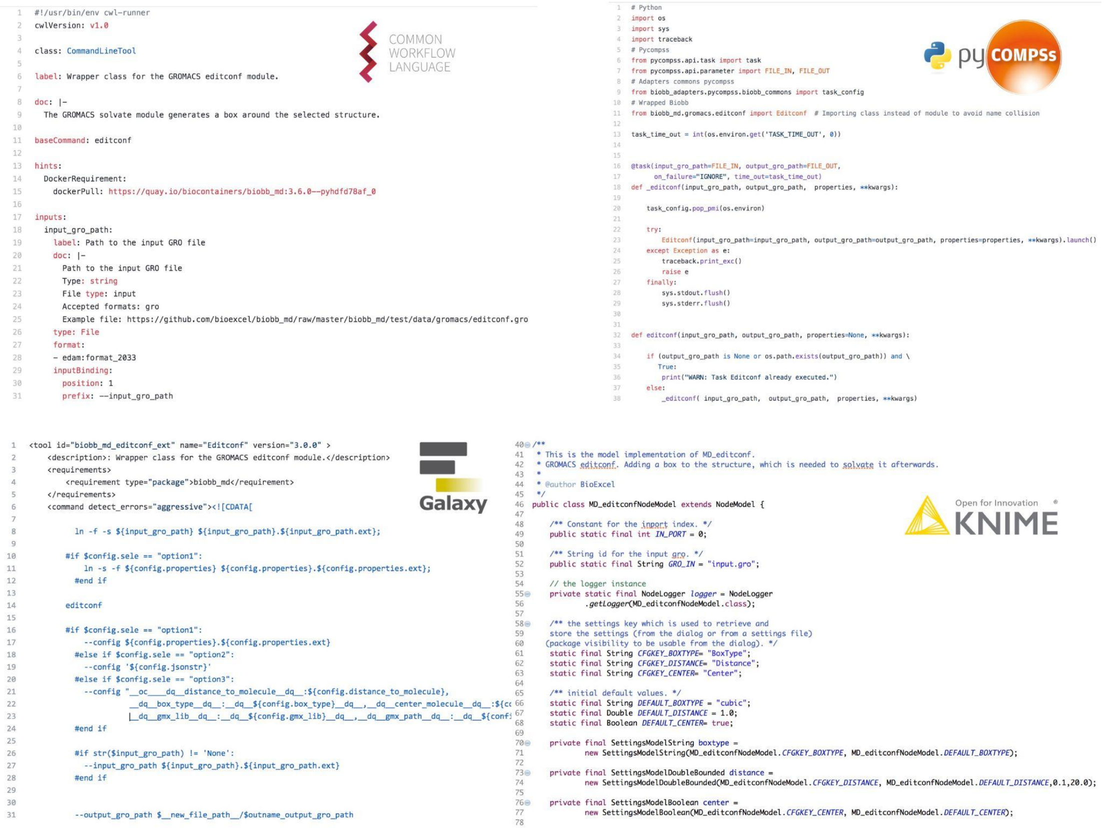
[17] https://toolshed.g2.bx.psu.edu/repository?repository_id=e23296b413014cfc

[18] https://github.com/bioexcel/biobb_adapters/tree/v0.1.4/biobb_adapters/pycompss

[19] https://github.com/bioexcel/biobb

[20] https://mmb.irbbarcelona.org/biobb-api/rest

[21] https://mmb.irbbarcelona.org/biobb-api/rest/swagger.json



```
#!/usr/bin/env cwl-runner
cwlVersion: v1.0

class: CommandLineTool

label: Wrapper class for the GROMACS editconf module.

doc: |-
  The GROMACS solvate module generates a box around the selected structure.

baseCommand: editconf

hints:
  DockerRequirement:
    dockerPull: https://quay.io/biocontainers/biobb_md:3.6.0--pyhdfd78af_0

inputs:
  input_gro_path:
    label: Path to the input GRO file
    doc: |-
      Path to the input GRO file
      Type: string
      File type: input
      Accepted formats: gro
      Example file: https://github.com/bioexcel/biobb_md/raw/master/biobb_md/test/data/gromacs/editconf.gro
    type: File
    format:
    - edam:format_2033
    inputBinding:
      position: 1
      prefix: --input_gro_path
```

```
# Python
import os
import sys
import traceback
# Pycompss
from pycompss.api.task import task
from pycompss.api.parameter import FILE_IN, FILE_OUT
# Adapters commons pycompss
from biobb_adapters.pycompss.biobb_commons import task_config
# Wrapped Biobb
from biobb_md.gromacs.editconf import Editconf  # Importing class instead of module to avoid name collision

task_time_out = int(os.environ.get('TASK_TIME_OUT', 0))


@task(input_gro_path=FILE_IN, output_gro_path=FILE_OUT,
      on_failure="IGNORE", time_out=task_time_out)
def _editconf(input_gro_path, output_gro_path,  properties, **kwargs):

    task_config.pop_pmi(os.environ)

    try:
        Editconf(input_gro_path=input_gro_path, output_gro_path=output_gro_path, properties=properties, **kwargs).launch()
    except Exception as e:
        traceback.print_exc()
        raise e
    finally:
        sys.stdout.flush()
        sys.stderr.flush()


def editconf(input_gro_path, output_gro_path, properties=None, **kwargs):

    if (output_gro_path is None or os.path.exists(output_gro_path)) and \
       True:
        print("WARN: Task Editconf already executed.")
    else:
        _editconf( input_gro_path,  output_gro_path,  properties, **kwargs)
```

```
<tool id="biobb_md_editconf_ext" name="Editconf" version="3.0.0" >
    <description>: Wrapper class for the GROMACS editconf module.</description>
    <requirements>
        <requirement type="package">biobb_md</requirement>
    </requirements>
    <command detect_errors="aggressive"><![CDATA[

          ln -f -s ${input_gro_path} ${input_gro_path}.${input_gro_path.ext};

        #if $config.sele == "option1":
            ln -s -f ${config.properties} ${config.properties}.${config.properties.ext};
          #end if

        editconf

        #if $config.sele == "option1":
            --config ${config.properties}.${config.properties.ext}
          #else if $config.sele == "option2":
            --config '${config.jsonstr}'
          #else if $config.sele == "option3":
            --config "__oc____dq__distance_to_molecule__dq__:${config.distance_to_molecule},
                     __dq__box_type__dq__:__dq__${config.box_type}__dq__,__dq__center_molecule__dq__:${cc
                     __dq__gmx_lib__dq__:__dq__${config.gmx_lib}__dq__,__dq__gmx_path__dq__:__dq__${confi
          #end if

          #if str($input_gro_path) != 'None':
            --input_gro_path ${input_gro_path}.${input_gro_path.ext}
          #end if


          --output_gro_path $__new_file_path__/$outname_output_gro_path
```

```
/**
 * This is the model implementation of MD_editconf.
 * GROMACS editconf. Adding a box to the structure, which is needed to solvate it afterwards.
 *
 * @author BioExcel
 */
public class MD_editconfNodeModel extends NodeModel {

    /** Constant for the inport index. */
    public static final int IN_PORT = 0;

    /** String id for the input gro. */
    public static final String GRO_IN = "input.gro";

    // the logger instance
    private static final NodeLogger logger = NodeLogger
            .getLogger(MD_editconfNodeModel.class);

    /** the settings key which is used to retrieve and
    store the settings (from the dialog or from a settings file)
   (package visibility to be usable from the dialog). */
    static final String CFGKEY_BOXTYPE= "BoxType";
    static final String CFGKEY_DISTANCE= "Distance";
    static final String CFGKEY_CENTER= "Center";

    /** initial default values. */
    static final String DEFAULT_BOXTYPE = "cubic";
    static final Double DEFAULT_DISTANCE = 1.0;
    static final Boolean DEFAULT_CENTER= true;

    private final SettingsModelString boxtype =
            new SettingsModelString(MD_editconfNodeModel.CFGKEY_BOXTYPE, MD_editconfNodeModel.DEFAULT_BOXTYPE);

    private final SettingsModelDoubleBounded distance =
            new SettingsModelDoubleBounded(MD_editconfNodeModel.CFGKEY_DISTANCE, MD_editconfNodeModel.DEFAULT_DISTANCE,0.1,20.0);

    private final SettingsModelBoolean center =
            new SettingsModelBoolean(MD_editconfNodeModel.CFGKEY_CENTER, MD_editconfNodeModel.DEFAULT_CENTER);

```

Figure 5.1: Code snippets for the BioBB WfMS bindings: CWL, PyCOMPSs, Galaxy and KNIME.

Since the publication of the library, several new building block modules[22] (Chemistry, Machine Learning, AMBER MD, Virtual Screening, etc.) have been added, and the set of operations for the existing BioBB families have been expanded. While BioExcel previously provided curated adapters (Figure 5.1) for running BioBB in workflow systems using Common Workflow Language (CWL) and PyCOMPSs, along with Galaxy Toolshed bindings, we have now started auto-generating these bindings, along with command line wrappers and REST web service APIs, using annotations within BioBB's Python docstrings as source. These annotations include sufficient information for a WfMS to launch a particular building block: *input* and *output* parameters (including *mandatory*/*optional* flags), compatible *formats* (including from EDAM ontology [Ison 2013]), *example* files (essential for testing purposes), default values and *dependencies*. This ensures human-readable documentation, FAIR metadata and programmatic accessibility can be generated consistently and comparably.

The library is showcased through a collection of demonstration workflows[23] [Hospital 2020]. Here, each workflow introduces individual building blocks as needed to explain a particular

[22] https://mmb.irbbarcelona.org/biobb/documentation/source

[23] http://mmb.irbbarcelona.org/biobb/workflows

scientific computational method. We primarily expose the workflows as Jupyter Notebooks [Kluyver 2016], which has been highlighted as a valuable tool for reproducible scientific workflows [Beg 2021]. This offers a graphical interactive interface, including documentation (integrated markdown) related to the workflow and the building blocks used, but also to the biomolecular simulation methods used in the pipeline. Moreover, as we have demonstrated with our own Binder[24] [Jupyter 2018] hosting, these workflows are reproducible across platforms, assisted by BioConda [Grüning 2018a] packaging of the building blocks and their software dependencies.

This assembly of available demonstration workflows have been successfully used in the BioExcel CoE for dissemination with a range of training events[25] (e.g. BioExcel Summer & Winter School, webinars and virtual training). In training we particularly utilised the Binder infrastructure of the BioExcel Cloud portal [Niewielska 2020] to give users a web-based first experience of the building blocks before they try them in other workflow systems.

We can observe that workflow building blocks such as BioBB are necessarily composed of a comprehensive list of digital objects, encompassing source code, packaging, containerization, documentation, attributions, citations, registry entries, WfMS integrations and REST APIs.

We propose to consider building blocks as *composite digital objects* in their own right: gathering the above software components along with their metadata, identifiers and operations then forms a *Canonical Workflow Building Block* (**CWBB**). We suggest this concept as a fundamental element of FAIR Digital Objects for Computational Workflows: researchers use the building blocks computationally as functional operations across WfMSs, while the FAIR aspect of CWBB propagates information and resources that are essential for reproducibility, reuse and understanding by anyone discovering the workflow.

5.1.2.1 Interoperability across different workflow languages

The concept of Canonical Workflow Building Blocks is here showcased with the BioBB library, by using a transversal workflow present in many different computational biomolecular projects: a Molecular Dynamics (MD) protein setup[26]. This workflow prepares a protein structure to be used as input for an MD simulation, going through a series of steps where the protein is completed (adding hydrogen and missing atoms), optionally introducing a residue mutation, then submerging the protein in a virtual box of water molecules with a particular ionic concentration, and finally energetically equilibrating the system (so that solvent and ions are well accommodated around the protein at the desired temperature).

This simulation process involves a non-negligible number of steps, using a variety of biomolecular tools. The BioBB library was used to assemble this workflow, interconnecting building blocks using Python functions (Jupyter Notebook, Command Line Interface), auto-generated bindings

[24] https://hub-bioexcel-binder.tsi.ebi.ac.uk/

[25] https://mmb.irbbarcelona.org/biobb/about/training

[26] http://mmb.irbbarcelona.org/biobb/workflows/tutorials/md_setup

(Galaxy [Afgan 2018], CWL [Crusoe 2022], PyCOMPSs [Tejedor 2017]) or manually generated bindings (KNIME [Fillbrunn 2017]). Corresponding workflows for the different WfMS can be found in WorkflowHub[27] [Lowe 2021a, Hospital 2021b, Bayarri 2021a, Bayarri 2021b, Hospital 2021a] and graphical extracts can be seen in Figure 5.2 on the next page.

This example demonstrates how the same canonical building blocks can be used in different WfMS. Wrappers and tools executed behind the workflows are exactly the same, but the workflows are built using different WfMS, some of them in a graphical way (drag & drop, Galaxy, KNIME), some in a command line way (Jupyter Notebook, PyCOMPSs, CWL); workflows can be focused on short/interactive executions (Jupyter Notebook), or on High Throughput/High Performance Computing (HT-HPC) executions (PyCOMPSs); some of them prepared for a particular WfMS installation (Galaxy), others completely system-agnostic (CWL).

The current number of available WfMS bindings include Jupyter Notebook, PyCOMPSs, CWL, Galaxy and KNIME WfMS, in addition to a command line[28] mechanism. Thanks to the extensive documentation added in the source code as Python docstrings, new bindings for available WfMS can be generated. We are also experimenting with generating a REST API exposing the building services as Web services. However, it should be noted that such automatic generation of bindings is not always practically feasible. As an example, KNIME nodes require a complete Java skeleton code, as well as a definition of new data types for all inputs/outputs required, which makes their automatic generation a heavy and potentially error-prone task. Bindings for workflow languages with a *domain-specific language* (DSL) for tool definitions (e.g. Galaxy, CWL) can on the other hand be generated in a more straightforward fashion.

The transversal protein MD setup workflow[29] was chosen as a real example that is readily understandable by domain experts. More complex pipelines[30] involving a broader set of wrapped biomolecular tools have been developed using the BioBB library, primarily as Jupyter Notebooks. A selection of these have similarly been assembled for different WfMS using the auto-generated bindings and uploaded to the WorkflowHub repository[31].

5.1.3 Discussion

Early work on libraries of workflows fragments include Web Service-based approaches where tools are wrapped and exposed using common, interoperable data types in BioMoby[32] [BioMoby 2008] for bioinformatics and similarly caBIG[33] [Saltz 2006] for cancer genomics. While these efforts were interoperable across WfMSs they required a large up-front investment in agreeing to and adapting native data to common RDF or XML representations.

[27] https://workflowhub.eu/collections/3

[28] http://mmb.irbbarcelona.org/biobb/availability/tutorials/command-line

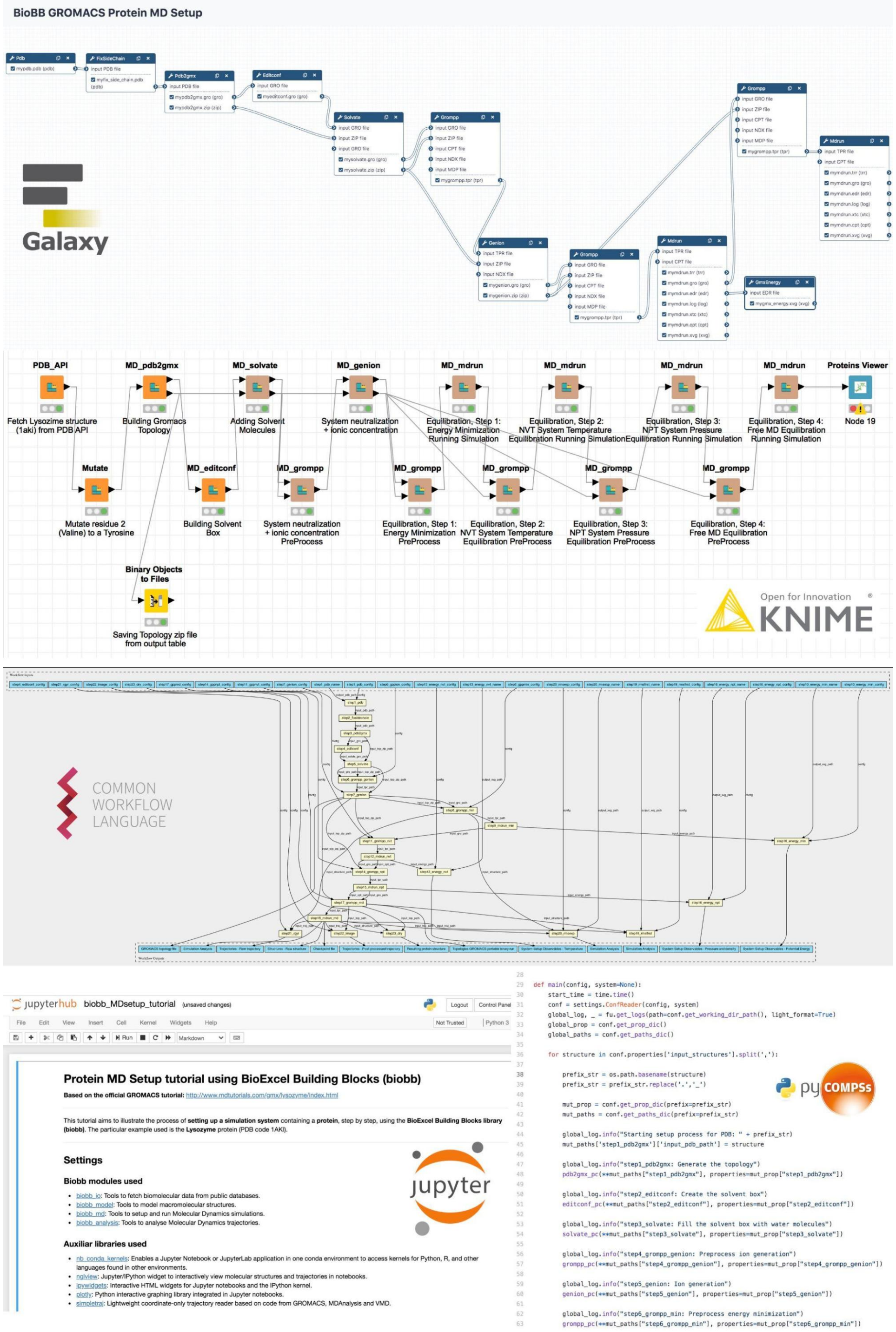
[29] https://workflowhub.eu/collections/3

[30] https://mmb.irbbarcelona.org/biobb/workflows

[31] https://workflowhub.eu/projects/11#workflows

[32] http://biomoby.open-bio.org/

[33] https://en.wikipedia.org/wiki/CaBIG


BioBB GROMACS Protein MD Setup
Pdb
FixSideChain
Pdb2gmx
Editconf
Solvate
Grompp
Genion
Grompp
Mdrun
Grompp
Mdrun
GmxEnergy
Galaxy
PDB_API
MD_pdb2gmx
MD_solvate
MD_genion
MD_mdrun
MD_mdrun
MD_mdrun
MD_mdrun
Proteins Viewer
Fetch Lysozime structure (1aki) from PDB API
Building Gromacs Topology
Adding Solvent Molecules
System neutralization + ionic concentration
Equilibration, Step 1: Energy Minimization Running Simulation
Equilibration, Step 2: NVT System Temperature Equilibration Running Simulation
Equilibration, Step 3: NPT System Pressure Equilibration Running Simulation
Equilibration, Step 4: Free MD Equilibration Running Simulation
Node 19
Mutate
MD_editconf
MD_grompp
MD_grompp
MD_grompp
MD_grompp
MD_grompp
Mutate residue 2 (Valine) to a Tyrosine
Building Solvent Box
System neutralization + ionic concentration PreProcess
Equilibration, Step 1: Energy Minimization PreProcess
Equilibration, Step 2: NVT System Temperature Equilibration PreProcess
Equilibration, Step 3: NPT System Pressure Equilibration PreProcess
Equilibration, Step 4: Free MD Equilibration PreProcess
Binary Objects to Files
Saving Topology zip file from output table
Open for Innovation KNIME
COMMON WORKFLOW LANGUAGE
step16_energy_npt
step19_rmsfirst
step20_rmsexp
step17_grompp_md
step18_mdrun_md
step21_rgyr
step22_image
step23_dry
GROMACS topology file
Simulation Analysis
Trajectories - Raw trajectory
Structures - Raw structure
Checkpoint file
Trajectories - Post-processed trajectory
Resulting protein structure
Topologies GROMACS portable binary run
System Setup Observables - Temperature
Simulation Analysis
Simulation Analysis
System Setup Observables - Pressure and density
System Setup Observables - Potential Energy
Workflow Outputs
jupyterhub
Logout
Control Panel
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Not Trusted
Python 3
Run
Markdown
jupyter
py COMPSs

biobb_MDsetup_tutorial (unsaved changes)

Protein MD Setup tutorial using BioExcel Building Blocks (biobb)

Based on the official GROMACS tutorial: http://www.mdtutorials.com/gmx/lysozyme/index.html

This tutorial aims to illustrate the process of **setting up a simulation system** containing a **protein**, step by step, using the **BioExcel Building Blocks library (biobb)**. The particular example used is the **Lysozyme** protein (PDB code 1AKI).

Settings

Biobb modules used

- biobb_io: Tools to fetch biomolecular data from public databases.
- biobb_model: Tools to model macromolecular structures.
- biobb_md: Tools to setup and run Molecular Dynamics simulations.
- biobb_analysis: Tools to analyse Molecular Dynamics trajectories.

Auxiliar libraries used

- nb_conda_kernels: Enables a Jupyter Notebook or JupyterLab application in one conda environment to access kernels for Python, R, and other languages found in other environments.
- nglview: Jupyter/IPython widget to interactively view molecular structures and trajectories in notebooks.
- ipywidgets: Interactive HTML widgets for Jupyter notebooks and the IPython kernel.
- plotly: Python interactive graphing library integrated in Jupyter notebooks.
- simpletraj: Lightweight coordinate-only trajectory reader based on code from GROMACS, MDAnalysis and VMD.

```python

def main(config, system=None):
    start_time = time.time()
    conf = settings.ConfReader(config, system)
    global_log, _ = fu.get_logs(path=conf.get_working_dir_path(), light_format=True)
    global_prop = conf.get_prop_dic()
    global_paths = conf.get_paths_dic()

    for structure in conf.properties['input_structures'].split(','):

        prefix_str = os.path.basename(structure)
        prefix_str = prefix_str.replace('.','_')

        mut_prop = conf.get_prop_dic(prefix=prefix_str)
        mut_paths = conf.get_paths_dic(prefix=prefix_str)

        global_log.info("Starting setup process for PDB: " + prefix_str)
        mut_paths['step1_pdb2gmx']['input_pdb_path'] = structure

        global_log.info("step1_pdb2gmx: Generate the topology")
        pdb2gmx_pc(**mut_paths["step1_pdb2gmx"], properties=mut_prop["step1_pdb2gmx"])

        global_log.info("step2_editconf: Create the solvent box")
        editconf_pc(**mut_paths["step2_editconf"], properties=mut_prop["step2_editconf"])

        global_log.info("step3_solvate: Fill the solvent box with water molecules")
        solvate_pc(**mut_paths["step3_solvate"], properties=mut_prop["step3_solvate"])

        global_log.info("step4_grompp_genion: Preprocess ion generation")
        grompp_pc(**mut_paths["step4_grompp_genion"], properties=mut_prop["step4_grompp_genion"])

        global_log.info("step5_genion: Ion generation")
        genion_pc(**mut_paths["step5_genion"], properties=mut_prop["step5_genion"])

        global_log.info("step6_grompp_min: Preprocess energy minimization")
        grompp_pc(**mut_paths["step6_grompp_min"], properties=mut_prop["step6_grompp_min"])
```

Figure 5.2: Protein MD Setup transversal workflow. Assembled in with 5 different workflow managers using BioBB canonical building blocks. From top-left: Galaxy [Lowe 2021a], KNIME [Hospital 2021b], CWL [Bayarri 2021a], Jupyter Notebook [Bayarri 2021b], PyCOMPSs [Hospital 2021a].

The notion of *abstract workflows* [Garijo 2011], structural workflow descriptions separated from their concrete execution realisations and augmented with Linked Data annotations, have been emphasised as essential for reuse and consistency across workflow systems. Identifying *common motifs* for workflow operations [Garijo 2014a] (e.g. Data preparation, Format transformation, Filter, Combine) are important to simplify and understand otherwise fine-grained workflow provenance traces.

Most other efforts to standardise a set of disparate analytical tools have been done within the scope of a single WfMS, allowing customised user interaction, data visualisation, configuration and findability, for instance Taverna components[34] had prototypical building blocks [De Giovanni 2016] which were instantiated at runtime by reference from a registry. KNIME components and metanodes[35], shared on the KNIME Hub[36] are frequently designed to be interoperable, but with a perhaps weaker notion of component families. The Galaxy toolshed[37] [Blankenberg 2014] is likewise populated with different sets of tool wrappers that are largely made to be interoperable within a category.

The Common Workflow Language (CWL) [Crusoe 2022] has a strong emphasis on interoperable command line tool descriptions, with support for containers[38] and Conda packaging, as well as support for FAIR metadata[39] like contributors, license and EDAM ontology type annotations. With multiple leading workflow engines now supporting CWL[40], and experimental Galaxy support, this seems perhaps the most promising candidate for both making and describing canonical workflow building blocks; however, we have identified a few stumbling blocks.

One obvious challenge is that the implementing WfMS needs to have CWL support, along with support for either containers or Conda packaging to find the described executables. While it is possible to run a CWL tool directly using a `#!/usr/bin/env cwl-runner` shebang[41] on POSIX systems, this still requires pre-installation and possibly configuration of a CWL engine like cwltool[42] or Toil[43] [Vivian 2017]. However workflow engines have multiple dependencies and often cannot easily be run from a container themselves[44].

Within the CWL community it was originally envisioned that a wider set of workflow systems would adopt CWL for tool description/execution, with a subset implementing full CWL workflow support. This would allow shared community effort for describing tools, say in the Common

[34] http://www.taverna.org.uk/documentation/taverna-2-x/components/

[35] https://docs.knime.com/2020-07/analytics_platform_components_guide/index.html

[36] https://hub.knime.com/

[37] https://toolshed.g2.bx.psu.edu/

[38] https://www.commonwl.org/user_guide/07-containers/

[39] https://www.commonwl.org/user_guide/17-metadata/

[40] https://www.commonwl.org/implementations/

[41] https://en.wikipedia.org/wiki/Shebang_(Unix)

[42] https://pypi.org/project/cwltool/

[43] https://toil.readthedocs.io/en/latest/running/cwl.html

[44] To execute the wrapped tool, a containerised workflow engine would need *nested containers* which are not generally recommended for security reasons. It is possible to work around this limitation using *Singularity* (https://sylabs.io/singularity/) or *Conda* (https://docs.bioexcel.eu/cwl-best-practice-guide/devpractice/containers/conda.html).

Workflow Library[45], rather than each WfMS needing to duplicate this tool wrapping in separate repositories and languages. However, with the exception of experimental tool support in Galaxy, in practice all CWL implementers have gone for full workflow support.

Another challenge is that making a set of building blocks frequently requires the use of *shims*, for instance file conversion, small search/replace operations or file renames. In a CWL approach these can either be performed with an Expression[46] using JavaScript snippets which only has limited access to file content, or as an additional workflow step added before or after the main tool step. This combination could then be nested as a subworkflow, similar to KNIME's *metanodes*, and would also be flexible by allowing different containers or packages for any pre- or post-steps. Such a CWL building block, however, becomes harder to access from a non-CWL WfMS, because of lack of control over configuration/execution options for the now nested CWL tools. In practice[47], executing a nested CWL workflow from a native WfMS language would require the engine to implement full CWL Workflow support (or delegate to a CWL engine).

For the main BioBB building blocks we implemented demonstrator workflows[48] that highlight how the tools should be used in different workflow management systems; each having a primary exemplar using Jupyter Notebook, which can be explored interactively using the BioExcel Binder[49]. If we consider the abstract demonstrator workflows as *canonical workflows* they are therefore very much active objects, but can also be seen as *workflow templates*, as any real use case will need to specialise the workflow to tweak parameters, data selection etc.

We therefore also provide such workflow templates for multiple WfMS, including CWL, PyCOMPSs and Galaxy. These are fairly disparate workflow languages, yet by the use of the same canonical workflow building blocks (which again invoke the same software binaries), such WfMS-specific workflows effectively are instantiations of the same canonical workflow.

One challenge found is how to publish such canonical workflows in registries like the WorkflowHub[50]. The hub supports the registration of Digital Objects in the form of RO-Crate [Soiland-Reyes 2022a], with the option of abstract CWL for describing the canonical workflow template, along with direct references to the workflow's GitHub repository.

[45] https://github.com/common-workflow-library/

[46] https://www.commonwl.org/v1.2/Workflow.html#Expressions_(Optional)

[47] It is worth mentioning that it would also be possible to generate WfMS-specific bindings from CWL descriptions (e.g. as demonstrated with cwl2script (https://github.com/common-workflow-lab/cwl2script) for Bash, gxargparse (https://github.com/common-workflow-lab/gxargparse) for Galaxy, cwl2wdl (https://github.com/common-workflow-lab/cwl2wdl) for WDL), although this necessitates constraining the tool and workflow definitions to a limited mappable subset of CWL.

[48] http://mmb.irbbarcelona.org/biobb/workflows

[49] https://hub-bioexcel-binder.tsi.ebi.ac.uk/h

[50] https://workflowhub.eu/

For instance in the RO-Crate for https://doi.org/10.48546/workflowhub.workflow.200.1 [Hospital 2021a], which can also be rendered[51] from GitHub, we have an entry for the *main workflow*[52] according to the Workflow RO-Crate profile[53], detailing each canonical workflow building block used (e.g. biobb-md metadata[54]). Here the FAIR aspect of the building blocks to help software citation is exercised, as the building block wrapper has one set of authors, documentation and licence (Apache-2.0), while the wrapped software (e.g. GROMACS metadata[55]) has different authors, licence (GPL-2.1+) and documentation.

However, the deposit of such RO-Crates in WorkflowHub results in one registration entry per workflow language, which are not otherwise related and may not even share the same source code repository. Thus, we have identified the need for adding an overall *canonical workflow entry*, which can bring in workflow documentation and references shared across WfMS implementations, including a set of links to the more granular canonical workflow building blocks used by the workflow, but also to the individual WfMS implementations as separate digital objects.

A similar question of granularity applies at the workflow tool level [Möller 2017], particularly for Findability and Accessibility, as we can consider at lowest granularity the *scientific method* in general (e.g. any algorithm for sequence alignment), followed by an *application suite* (bio.tools entry [Ison 2021], homepage, documentation), instantiated as a particular *software installation* (Debian package, Docker container) with its dependencies at same level. The installation includes one or more *software executables* (a particular binary, a running service service), providing at the highest detailed granularity level the specific types of *software functionality* (a particular mode of operation, choice of analysis), for instance using certain command line flags.

For canonical workflow building blocks, with a focus on pluggable composability, this is mainly defined at this high granularity level of specific software functionality: explicit operations from an installed tool, which are then combined in a workflow. This is indeed the level WfMS tool definitions are typically done, e.g. a CWL Command Line Tool specifies a particular way to run a particular software binary. However, to be an actionable CWBB, the building block needs to additionally convey the lower granularity levels; particularly to support multiple options for interoperable installation and execution, as well as metadata at the most general level, such as documentation and scholarly citations.

While workflow management systems typically only operate at the highest granularity levels for execution details, and are frequently unaware of (or not exposing metadata at) the more general levels, we argue that in order for a Canonical Workflow [CWFR 2021] to follow and support

[51] https://rawcdn.githack.com/bioexcel/biobb_hpc_workflows/53958e7c278e53c277a7217057b785482f193f7f/ro-crate-preview.html

[52] https://rawcdn.githack.com/bioexcel/biobb_hpc_workflows/53958e7c278e53c277a7217057b785482f193f7f/ro-crate-preview.html#workflows/MD/md_list.py

[53] https://w3id.org/workflowhub/workflow-ro-crate/1.0

[54] https://rawcdn.githack.com/bioexcel/biobb_hpc_workflows/53958e7c278e53c277a7217057b785482f193f7f/ro-crate-preview.html#https%3A//pypi.org/project/biobb-md/3.6.0/

[55] https://rawcdn.githack.com/bioexcel/biobb_hpc_workflows/53958e7c278e53c277a7217057b785482f193f7f/ro-crate-preview.html#https%3A//doi.org/10.5281/zenodo.2564764

FAIR principles for itself and its data, the workflow management system need to *propagate structured metadata* about the tools used by the workflow. We propose that in order to support the workflow's applicability to multiple WfMS, the tools themselves must also have a consistent packaging and formal description that enables consistent computational invocation.

At the most general level, a canonical workflow built using such CWBBs is even conceptually reproducible because the FAIR documentation of the workflow, through its canonical workflow building blocks, identifies how individual tools and software applications are composed, which in worst case can be rebuilt using different installation methods in a different WfMS, or in best case inspected to detect and cross-link the same canonical workflow appearing in different WfMS instantiations. This view of software as composition of other software typically also applies at individual tool level, which themselves depend on programming language runtimes, libraries, services and reference data.

5.1.4 Requirements for Canonical Workflow Building Blocks

Building on the experiences with BioBB, we here propose requirements and recommendations for establishing Canonical Workflow Building Blocks (CWBB) as implementations of *canonical steps* introduced for Canonical Workflow Frameworks for Research [CWFR 2021].

The core purpose of a CWBB is to wrap a command line tool or other software that can perform an operation as part of a computational workflow. As such, the general advice for making software workflow-ready applies [Brack 2022a] (e.g. easy to install, documented, parallelizable, reproducible output); however, a CWBB is also permitted to make use of additional scripts or *shims* to further adapt a third-party tool for workflow use and for data interoperability across blocks.

The way tools are installed or invoked varies slightly across WfMS and operating systems, therefore a CWBB should provide multiple methods for distributing software; currently containers (Docker, Singularity) and distribution-independent packaging (e.g. Conda, Homebrew) are promising by having reproducible install recipes and a wide range of open source dependencies (e.g. Java, Python). Additionally building blocks should allow overriding execution paths, e.g. for use with HPC module system and hardware-optimised binaries.

The CWBBs should have sufficient annotations to be able to generate bindings for different WfMSs and REST APIs, e.g. parameter names and descriptions, types and default values; enumerators for options, file formats for inputs/outputs.

Building blocks should be grouped into families that are interoperable through common data structures and file formats, as well as having joint naming conventions for configuration options. A CWBB family should be released as a single version following semantic versioning[56] rules, which should have a corresponding persistent identifier (PID) [McMurry 2017].

Metadata for CWBBs should be captured following FAIR guidelines, and distributed as part

[56] https://semver.org/spec/v2.0.0.html

of the block family and resolvable from the PID as a FAIR Digital Object. Metadata should include references to the CWBB software distributions (e.g. quay.io[57] container URL) as well as attributions, citations and documentation for the wrapped tool.

Example workflows showing CWBB usage should be included in a WfMS-neutral language such as Jupyter Notebooks, which may have equivalent variants for each workflow binding. These workflows should be registered in a workflow registry like WorkflowHub or Dockstore, and assigned their own PIDs.

5.1.5 Conclusions

The proposed concept of Canonical Workflow Building Blocks can bridge the gap between FAIR Computational Workflows, interoperable reproducibility and for building canonical workflow descriptions to be used and described FAIRly across WfMSs.

The realisation of CWBBs can be achieved in many ways, not necessarily using the Python programming language together with RO-Crate as explored here. In particular if the envisioned Canonical Workflow Frameworks for Research become established in multiple WfMSs with the use of FAIR Digital Objects, the different implementations will need to agree on object types, software packaging and metadata formats in order to reuse tools and provide interoperable reproducibility for canonical workflows.

Likewise, to build a meaningful collection of building blocks for a given research domain, a directed collaborative effort is needed to consistently wrap tools for a related set of WfMSs, chosen to target particular use cases (a family of canonical workflows).

For individual users, a library of Canonical Workflow Building Blocks simplifies many aspects of building pipelines, beyond the FAIR aspects and data compatibility across blocks. For instance, they can benefit from training of a CWBB family using Jupyter Notebooks, and then use this knowledge to utilise the same building blocks in a scalable HPC workflow with a CWL engine like Toil, knowing they will perform consistently thanks to the use of containers.

While we have demonstrated CWBB in the biomedical domain, this approach is generally applicable to a wide range of sciences that execute pipelines of multiple file-based command line tools—however, it may be harder to achieve with more algebraic "in memory" types of computational workflows, where steps could be challenging to containerize and distinguish as separate block.

We admit that biomolecular research is quite a homogenous field with respect to computational analyses and now becoming relatively mature in terms of tool composability in workflows, building on the experiences of the "FAIR pioneers" in the field of bioinformatics. Other fields, such as social sciences or ecology, can have a wider variety of methods and computational tools, often with human interactions, and may have to adapt the software to be workflow-ready [Brack 2022a] before using them as Canonical Workflow Building Blocks. Domains

[57] https://quay.io/search

adapting CWBB approach (or workflow systems in general) should take note of the great benefits of hosting collaborative events where developers meet each other and their potential users, demonstrated in our field with events such WorkflowsRI [Ferreira da Silva 2021] and Biohackathons [Garcia 2020b].

The Common Workflow Language shows promise as a general canonical workflow building blocks mechanism: gathering execution details of tools along with their metadata and references, augmented with abstract workflows[58] to represent canonical workflows. However, this would need further work to implement our CWBB recommendations in full. Future work for the Canonical Workflow Building Blocks concept includes formalising and automating publication practises, to make individual blocks available as FAIR Digital Objects on their own or as part of an aggregate collection like RO-Crate.

[58] https://docs.bioexcel.eu/cwl-best-practice-guide/devpractice/partial.html#using-abstract-operations-as-placeholders

5.2 The Specimen Data Refinery

A canonical workflow framework and FAIR Digital Object approach to speeding up digital mobilisation of natural history collections

A key limiting factor in organising and using information from physical specimens curated in natural science collections is making that information computable, with institutional digitization tending to focus more on imaging the specimens themselves than on efficiently capturing computable data about them. Label data are traditionally manually transcribed today with high cost and low throughput, rendering such a task constrained for many collection-holding institutions at current funding levels.

We show how computer vision, optical character recognition, handwriting recognition, named entity recognition and language translation technologies can be implemented into canonical workflow component libraries with findable, accessible, interoperable, and reusable (FAIR) characteristics. These libraries are being developed in a cloud-based workflow platform—the 'Specimen Data Refinery' (SDR)—founded on Galaxy workflow engine, Common Workflow Language, Research Object Crates (RO-Crates) and WorkflowHub technologies. The SDR can be applied to specimens' labels and other artefacts, offering the prospect of greatly accelerated and more accurate data capture in computable form.

Two kinds of FAIR Digital Object (FDO) are created by packaging outputs of SDR workflows and workflow components as digital objects with metadata, a persistent identifier, and a specific type definition. The first kind of FDO are computable Digital Specimen (DS) objects that can be consumed/produced by workflows, and other applications. A single DS is the input data structure submitted to a workflow that is modified by each workflow component in turn to produce a refined DS at the end. The Specimen Data Refinery provides a library of such components that can be used individually, or in series. To cofunction, each library component describes the fields it requires from the DS and the fields it will in turn populate or enrich. The second kind of FDO, RO-Crates gather and archive the diverse set of digital and real-world resources, configurations, and actions (the provenance) contributing to a unit of research work, allowing that work to be faithfully recorded and reproduced.

Here we describe the Specimen Data Refinery with its motivating requirements, focusing on what is essential in the creation of canonical workflow component libraries and its conformance with the requirements of an emerging FDO Core Specification being developed by the FDO Forum.

5.2.1 Introduction

A key limiting factor in organising and using information from physical specimens curated in natural history collections is making that information computable ('machine-actionable') and extendable. More than 85% of available information currently resides on labels attached to specimens or in physical ledgers [Walton 2020a]. Label data are commonly transcribed manually with high cost and low throughput, rendering such a task constraining for many institutions at current funding levels. However, the advent of rapid, high-quality digital imaging has meant that digitizing specimens, including their labels, is now faster and cheaper [Thiers 2016]. With initiatives such as Advancing Digitization of Biological Collections (ADBC), integrated Digitized Biocollections (iDigBio) and the Distributed System of Scientific Collections (DiSSCo) [Nelson 2019a, Nelson 2019b, Addink 2019, Lannom 2020] aiming to increase the rate and accuracy of both mass and on-demand digitization of natural history collections, the gap between expectations of what should be digitally available and computable, and what can be achieved using traditional transcription approaches is widening. Modern, highly efficient workflow tools and approaches can play a role to address this.

Collection digitization began towards the end of the 20th century by typing basic data from labels into the collection (asset) management systems of collection-holding institutions such as natural history museums, herbaria and universities. Initially, this was to facilitate indexing and cataloguing and locating the physical specimens, but with the addition of photographic images of specimens and the public availability of specimen data records, through data portals of the institutions themselves as well as international data infrastructures like the Global Biodiversity Information Facility (GBIF), such bodies of data have been rapidly exploited for research [GBIF 2021, Heberling 2021]. It has become clear that widespread digitization of data about physical specimens in collections and the advent of high-throughput digitization processes [Sweeney 2018, Allan 2019, Hereld 2019, Price 2018, Tegelberg 2017] is transforming and will radically further transform the range of scientific research opportunities and questions that can be addressed [Heberling 2019, Kharouba 2019]. Scientific conclusions and policy decisions evidenced by digital specimen data enhance humankind's ability to conserve, protect, and predict the biodiversity of our world [Watanabe 2019, Lughadha 2019].

Harnessing technologies developed to harvest, organise, analyse and enhance information from sources such as scholarly literature, third-party databases, data aggregators, data linkage services and geocoders and reapplying these approaches to specimens' labels and other artefacts offers the prospect of greatly accelerated data capture in a computable form [Owen 2020]. Tools of particular interest span the fields of computer vision, optical character recognition, handwriting recognition, named entity recognition and language translation.

Workflow technologies from the ELIXIR Research Infrastructure [Harrow 2021], including Galaxy [Afgan 2018], Common Workflow Language [Crusoe 2022], Research Object Crates (RO-Crates) [Ó Carragáin 2019a, Soiland-Reyes 2022a] and WorkflowHub [Goble 2021], and selected tools are integrated in a cloud-based workflow platform for natural history specimens he 'Specimen Data Refinery' [Walton 2020a] that will become one of the main services to be offered by the planned DiSSCo research infrastructure [Addink 2019]. The tools themselves, implemented with findable, accessible, interoperable, and reusable (FAIR) characteristics [Wilkinson 2016] are packaged into canonical workflow component libraries [Wittenburg 2022b], rendering them reusable, and interoperable with one another. FAIR Digital Objects are adopted as the common input/output pattern, fully compatible with digital objects at the core of DiSSCo data management [Hardisty 2019b].

The Refinery brings together domain-specific workflows for processing specimen images and extracting text and data from images with canonical forms for components and interactions between components that can lead to improved FAIR characteristics for both the workflows themselves and the data resulting from workflow execution.

FAIR Digital Objects (FDOs) are created by packaging outputs of workflows and workflow components as digital objects with metadata, a persistent identifier, and a specific type definition against which operations can be executed [De Smedt 2020]. The Refinery uses two kinds of FDOs:

- **computable Digital Specimen (DS) objects** [Hardisty 2020] from DISSCo for the scientific input/output data that can be consumed/produced by workflows and other applications.
- **workflow objects, implemented as RO-Crates** [Soiland-Reyes 2022a], from ELIXIR gather and archive the diverse set of workflow process data—the digital and real-world resources, configurations and actions (the provenance) contributing to a unit of digitization or other work producing the Digital Specimen digital objects, allowing that work to be scrutinised and faithfully reproduced if necessary.

We first summarise related work before describing the problem to be addressed by the Specimen Data Refinery. We then explain our Canonical Workflows for Research (CWFR) approach using these FDOs in the design of the SDR, the experimental setup, and results so far from the work in progress. While future work will clarify full results and challenges of implementing a robust, reliable, and easy-to-use production-capability SDR, in this early report following SDR prototyping and conceptualization, we focus on what we found to be essential in the use of FDOs and CWFR canonical step libraries, and on the compliance of canonical workflow (component) inputs and outputs with the requirements of the FDO Framework [Bonino 2019].

5.2.2 Related Work

5.2.2.1 Workflows for processing specimen images and extracting data

While natural history collections are heterogeneous in size and shape, often they are mass digitized using standardised workflows [Sweeney 2018, Allan 2019, Hereld 2019, Price 2018, Tegelberg 2017]. In pursuit of higher throughput at lower cost, yet with higher accuracy and richer metadata, further automation will increasingly rely on techniques of object detection and segmentation, optical character recognition and semantic processing of labels, and automated taxonomic identification and visual feature analysis [Walton 2020a, Owen 2020].

Although there is a great deal of variety among images of different kinds of collection objects that are digitized (see Figure 5.3 on the following page) there are visual similarities between them. Most images contain labels, scale bars and often, colour charts as well as the specimen itself. This makes them amenable to improved approaches to object detection [Triki 2020] and segmentation

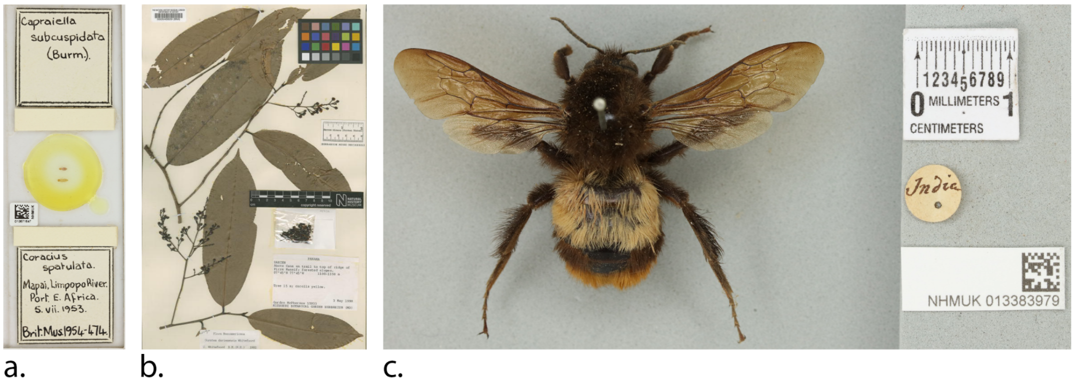


Figure 5.3: A range of specimen images. From the Natural History Museum, London, demonstrating the diversity of collection objects, which include handwritten, typed, and printed labels. (a) Microscope slide (NHMUK010671647)[59], (b) Herbarium specimen (BM000546829)[60], (c) pinned insect (NHMUK013383979)[61].

into 'regions of interest' [Nieva de la Hidalga 2021] as precursive steps for multiple kinds of workflows.

Segmentation, specifically, can be employed as an early step in a workflow to send just the relevant region(s) of interest from an image to later workflow steps. Not only does this decrease data transfer time and minimise computational overheads but it can also substantially increase the accuracy of subsequent OCR processing and semantic recognition steps [Owen 2020].

Much of the data about specimens is stored on their handwritten, typed or printed labels or in registers/ledgers [Walton 2020b]. Direct manual transcription into local databases with manual georeferencing is the primary method used today to capture this data. Potentially, OCR can significantly increase transcription speeds whilst reducing cost; although it sacrifices accuracy and disambiguation that are today achieved with specialist knowledge provided by humans during the process. Returning character strings from OCR is useful, but semantically placing this data in its context as information specific to natural history specimens and linking that back to the original physical specimens is of much higher value, improving the utility of natural history collections. Shortfalls in accuracy and disambiguation can be made up by exploiting Natural Language Processing (NLP) advances such as named entity recognition to identify text segments belonging to predefined categories (for example, species name, collector, locality, date) [Owen 2020]. Nevertheless, this only works well on a small proportion of captured data in the absence of 'human in the loop' input. To better automate disambiguation of people's names, for example, access to other contextual 'helper' data are needed (e.g., biographical data in Wikidata) as well as cross-comparison with other data from the specimen, such as the date of

[59] https://data.nhm.ac.uk/object/c65d9a3c-d8f6-4fac-a418-05c3b697cece
[60] https://data.nhm.ac.uk/object/be595f07-73c5-4764-a96c-8b377e3d1507
[61] https://data.nhm.ac.uk/object/745febc7-8222-498a-9969-5f6b12f85ef3

collection and location [Groom 2020].

Automated identification of species from images of living organisms has achieved impressive levels of accuracy [Knyshov 2021, Hussein 2021, Carranza-Rojas 2017, Little 2020, Pryer 2022, Unger 2016] with techniques translated to an increasing range of enthusiastically received consumer applications for plant and animal identification using mobile phones (e.g., Plantsnap[62], PictureThis[63], iNaturalist SEEK[64]). Automated identification of *preserved specimens*, however, presents different challenges. Although identification might be made more accurately because a specimen is presented in a standard manner, separated from other organisms and the complexity of a natural background, the loss of colour and distortion of the shape of the organism arising from preparation and preservation processes can lead to the loss of important identification clues that might be present on a living example.

5.2.2.2 Workflow management systems and canonical workflows for research

A workflow chains together atomised and executable components with the relationships between them to clearly define a control flow and a data flow. Their significant defining characteristics are (i) abstraction, through the separation of the workflow specification (the work to be done) from its execution (how it is done), and (ii) composition whereby the components can be cleanly combined and reused and workflows themselves can be neatly packaged as components [Atkinson 2017]. Workflow Management Systems (WfMSs) typically provide the necessary mechanisms for explicitly defining workflows in a reusable way together with a workflow engine that executes the workflow steps and keeps an accountable record of the processing—logging the codes executed and the data lineage of the results. In the past decade there has been a rise in popularity in both the development of WfMS and their use, driven by the increasing scales of data and the accompanying complexity of its processing [Atkinson 2017].

Workflow management systems typically vary in the features they provide for supporting: workflow programming language and control flow expressivity; data type management; code wrapping, containerisation and integration with software management tools; exploitation of computational architectures; availability of development and logging tools; licensing and so on. Although several hundred kinds of such systems exist [Amstutz 2021], communities tend to cluster around a few popular systems based on their "plugged-in" availability of data type specialist codes, the catered-for skills level of the workflow developers, and its documentation, community support and perceived sustainability. For the Specimen Data Refinery, the Galaxy workflow system [Afgan 2018] in conjunction with Common Workflow Language (CWL) [Crusoe 2022] has been chosen. CWL is a workflow specification standard geared towards supporting interoperable and scalable production pipelines, abstracting away from the internal data structures of some of the language-specific workflow systems.

Originally designed for computational biology and with many available tool components, Galaxy

[62] https://www.plantsnap.com/
[63] https://www.picturethisai.com/
[64] https://www.inaturalist.org/pages/seek_app

[Afgan 2018] supports multiple domains. Workflows can be built by manually experimenting with data manipulations in a 'data playground' and subsequently converting histories of those to workflows, or by a more traditional drag-and-drop composition approach. New components can be created by wrapping existing programs, with in-built dependency management and automated conversion to executable containers. As such, Galaxy and CWL offer possibilities for a rich canonical workflow component landscape with a workflow management regime that can be both easily FAIR compliant and efficient internally [Wittenburg 2022b]. The WorkflowHub, which facilitates CWL and enables workflows to be registered, shared and published, is mutually coupled with Galaxy so that workflows can be discovered in the Hub and immediately executed in a public-use Galaxy instance.

In the context of the SDR, users can construct institution or project-specific variants of digitization workflows to suit their specific needs. As collections are heterogeneous, different specimen types or specific sets of specimens are likely to have variations and idiosyncrasies in the digitization and processing needed. Tools for automated identification of specimens are likely to be taxon-specific, and as such it seems likely that taxon-specific workflows will become common. In addition, institutions have specific data exchange requirements for their individual collection management systems. Ensuring that workflows can be easily modified in a common environment bridges the gap between community contribution to shared tooling and the bespoke needs of specific institutions/collections.

Although computational workflows typically emphasise scalable automated processing, in practice many also combine automation with manual steps. This feature is also supported by Galaxy and CWL, allowing (for example) manual geocoding and verification during the digitization process of the locations where specimens were collected.

5.2.2.3 FAIR Digital Objects

Galaxy/CWL environments offer the possibility to integrate generic digital object methods [Hui 2012, Kallinikos 2013, Kahn 2006] for the interactions between workflow components, thus making them able to meet the need and ease the burden of compiling FAIR compliant data throughout the research lifecycle [Wittenburg 2022b].

A digital object exhibiting FAIR characteristics is a FAIR Digital Object [De Smedt 2020] and is defined formally as "a unit composed of data and/or metadata regulated by structures or schemas, and with an assigned globally unique and Persistent Identifier (PID), which is findable, accessible, interoperable and reusable both by humans and computers for the reliable interpretation and processing of the data represented by the object".

Supporting 'FAIRness' internally and acting as glue between the steps of canonical workflows, FDOs record and can represent the state of a workflow, its inputs and outputs, and the component steps performed in a comprehensive manner [Wittenburg 2022b]. Each FDO is anchored by a globally unique and resolvable PID (such as a DOI®, for example) that clearly refers to one digital entity. The PID resolution offers persistent references to find, access and reuse all information entities that are relevant to access and interpret the content of an FDO. In doing so, the FDO creates a new kind of machine-actionable, meaningful and technology independent unit of information. This is both immediately available and amenable for further use, as well as being comparable to the role of the classical archival storage box when necessary.

Computable Digital Specimens as a kind of FAIR Digital Object Digital Specimens (DS) are a specific class of FDO that group, manage and allow processing of fragments of information relating to physical natural history specimens. On a one-to-one correspondence a DS authoritatively collates data about a physical specimen (i.e., information extracted and captured from labels by digitization workflows) with other data—often to be found from third-party sources – derived from analysis and use of the specimen.

openDS [openDS 2021] is the developing specification for open Digital Specimens and other related object types, defining:

- The logical structure and content of Digital Specimen (DS), Basic Image Object (BIO) and other object types, and the operations permitted on them.
- The handling rules and behaviors governing digital specimen object operations in general.
- Serialization and packaging as JavaScript Object Notation (JSON) for lightweight data interchange between systems, sub-systems and components of systems (for which, read 'workflow components' [Bray 2017].

openDS is essential to future FAIR digitization of natural history collections and to Digital Specimens as self-standing digital objects on the Internet, amenable to computer processing. It contributes to the new transformative generation of FAIR infrastructure and applications based on Digital Object Architecture that is planned for the Distributed System of Scientific Collections (DiSSCo) [Lannom 2020, Addink 2019, Hardisty 2020] European research infrastructure.

Henceforth we refer to these as **openDS FDOs**.

FAIR packaging of research/workflow objects with RO-Crate The useful outcomes of research are not just traditional publications nor data products but everything that goes into and supports an investigative work or production pipelining activity. This includes input and intermediate data, parameter settings, final outputs, software programs and workflows, and configuration information sufficient to make the work reproducible. Research objects [Bechhofer 2013] are a general approach to describing and associating all of this content together in a machine-readable form so that it can be easily preserved, shared and exchanged. Workflow objects are a specific subclass of research objects.

RO-Crate[65] [Ó Carragáin 2019a, Soiland-Reyes 2022a] has been established as a community standard to practically achieve FAIR packaging of research objects with their structured metadata. Based on well-established Web standards, RO-Crate uses JSON-LD [Sporny 2020]

[65] Section 4.1 on page 77

with [schema.org] for its common metadata representation. It is extensible with domain-specific vocabularies in a growing range of specializing RO-Crate profiles, e.g., for domains such as earth sciences [Corcho 2021], biosciences [Goble 2021]; for object types such as data or workflow [Bacall 2022]; or for workflow runs). RO-Crate has been proposed for the implementation of FAIR Digital Objects on the World Wide Web as a common representation of the FDO Metadata objects foreseen by the FDO Framework [Goble 2021, Bonino 2019]. Combined with FAIR Signposting [Van de Sompel 2022] for resolving Persistent Identifiers to FDOs on the World Wide Web, these RO-Crates are findable, accessible, interoperable, and reusable by machines to both create and obtain the information they need to function.

Henceforth, we refer to **RO-Crate FDOs**.

5.2.3 Problem Description

5.2.3.1 Automating digitization and capturing the process

In the lengthy history of collectors and museums curating artefacts and specimens, we see that there have been and always will be ambiguities, uncertainties, and inaccuracies in interpretations of recorded information and attached labels [Lohonya 2020]. The practices of different collectors and curators vary and change over time. There are constraints of the label medium itself arising from the specifics of accepted preparation and preservation processes (e.g., tiny, handwritten labels pinned to butterflies).

Although systematic digitization of label and other recorded data can help to unify otherwise diverse information (e.g., species names, locations) the digital process and the resulting digital specimen data carry their own assumptions, simplifications, inconsistencies, and limitations. Over time, tools and methods, workflows and data models all evolve and improve. In particular, increasing automation for throughput and accuracy often involves increased assistance from computers and software.

Just as manual curation and improvement work implies the need for good record keeping, so too does working digitally imply the importance of ensuring that sufficient records are captured about the computer-assisted digitization and curation processes (provenance). These justify the produced digital specimen data and propagate credit for work done to their analogue equivalents, and also allow retrospective review, revision or recomputation of the produced data as future needs, practices or knowledge change.

Globally, there is underinvestment and missing technical expertise for wide-scale automated mass digitization. Sharing proven digitization workflows via a repository or registry linked to an individual published journal article presents significant barriers to re-use. Exploiting hosted community environments—in this case Galaxy and WorkflowHub—for the deployed tooling lowers barriers and provides rapid and easy access for institutions with limited capabilities and capacities for digitization. Hosted workflows represent "primacy of method" for a community evolving towards a new research culture that is becoming increasingly dependent on working digitally and collaboratively [De Roure 2010, Hardisty 2016].

5.2.3.2 Users, user stories and specimen categories

Initially, two kinds of users must be supported: digitizer technicians and collections managers/curators. Five high-level user stories describe and broadly encompass the functionality these users need:

1. As a digitizer, I want to construct a workflow from a set of predefined components, so I can use that workflow to digitize specimens to a predefined specification.
2. As a digitizer, I want to run one or many specimen images through a workflow so I can create new digital specimens.
3. As a collection manager/curator, I want to run one or many digital specimens through a workflow to enrich my digital specimens with further data.
4. As a collection manager/curator, I want to view the metadata of a digitization workflow run so I can understand what happened on that run.
5. As a digitizer, I want to export the output of a digitization run, so I can consume the output of a digitization run into my institution's collection management system.

To prove the SDR concept, three categories of preserved specimen types have been selected to be supported initially: herbarium sheets, microscope slides and pinned insects (see Figure 5.3 on page 136).

5.2.4 The FDO and CWFR approach in the Specimen Data Refinery

Workflows will be designed to support the user categories and stories given above. The performance of the SDR will be evaluated against these specimen types, eventually using several thousand different specimen and label images. This is in anticipation of SDR becoming part of the pivotal technology to achieve high rates of mass FAIR digitization expected through the planned DiSSCo research infrastructure [Lannom 2020, Addink 2019, Hardisty 2020].

5.2.4.1 FDO types

In the Specimen Data Refinery (see Figure 5.4 on the next page) the role of openDS FDOs is planned as the basis for the primary workflow inputs and outputs, and for data transfer and interactions between components within SDR workflows. A single openDS FDO submitted to the beginning of the workflow (or a *de novo* digitization that is immediately wrapped as a new openDS FDO) becomes modified by each workflow component to produce an incrementally refined openDS FDO. FDOs are acting as the unit of data communication between canonical workflow components, in that each step is immediately creating an FDO with associated FAIR compliant documentation.

RO-Crate FDOs capture two aspects of a workflow:

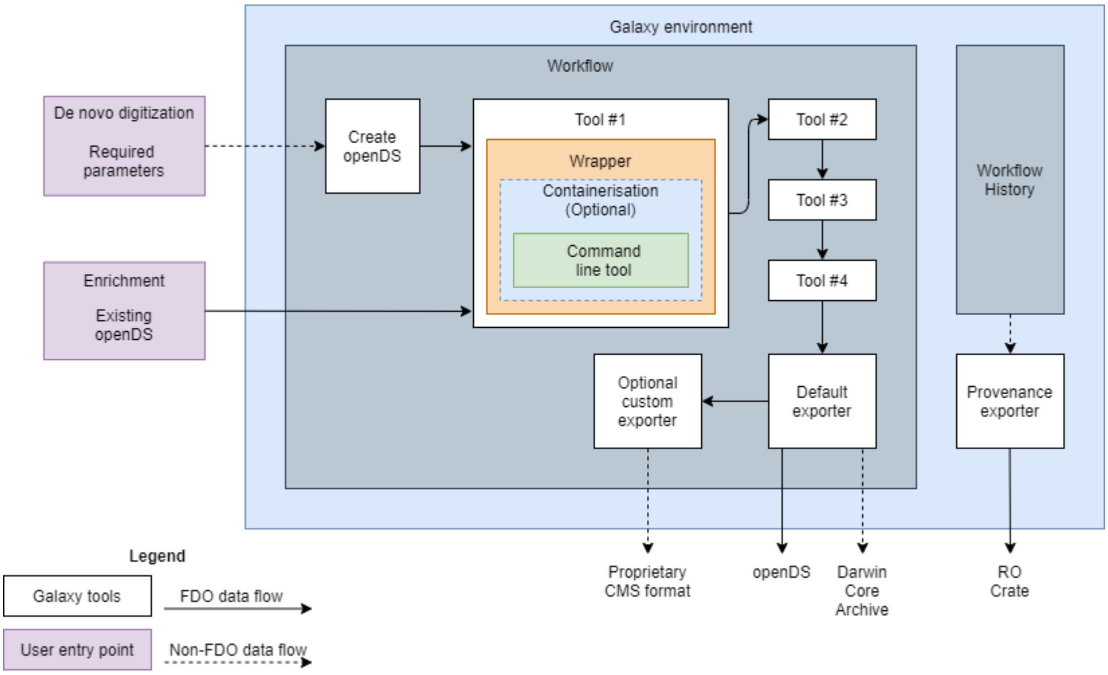


Figure 5.4: CWFR approach. Adopted for the SDR as a Galaxy workflow management system implementation with '*de novo* digitization' and 'enrichment' entry points.

1. A **Workflow-RO-Crate** contains the workflow definition, the computational tools and configuration, graphical image of the workflow, etc; this is the *method* registered in the WorkflowHub and activated in the Specimen Data Refinery for execution.

2. A **Workflow-Run-RO-Crate** references (a) and records the details of a specific computational workflow run and its runtime information, with relations to the used and generated FDOs. This captures the digitization provenance that is generated as the openDS FDO makes its journey through a workflow.

The final step in SDR workflows can be a data exporter tool, allowing users to export the entire openDS object as is, or to convert and export it in another format, such as CSV, DarwinCore Archive, etc. This reconfigurable nature of data export allows users to define their own transformer function to allow export to match formats specific to specific collection management systems in use by their institution, such that refined data can be repatriated. The provenance exporter transforms Galaxy workflow history data into a Workflow-Run-RO-Crate FDO.

All FDO types are serialised as JSON.

5.2.4.2 Canonical components

Each workflow component from a canonical library (to be built, illustrated as tools #1 - #4 in Figure 5.4) describes what attributes it requires from the openDS FDO to be able to function, and the attributes it will in turn populate or enrich. The interface between the component and the openDS FDO is formed by the wrapper (orange in Figure 5.4) around the (optionally) containerised command-line tool (blue/green in figure 5.4). Canonical components can be used individually, or in series. The openDS FDO data flows between components will always be of the same type, being modified as the workflow proceeds.

This allows tools to function both as standalone components and as part of any sequence of chained tools, provided that the specific openDS FDO attributes required for a tool to work are pre-populated. This keeps the SDR flexible and customisable for different digitization pipelines.

Two entry points are provided for users of the SDR. One is named as the '*de novo* digitization' entry point (Figure 5.4) fulfilling the needs of user story (2) where specimens are being newly digitized for the first time. The second entry point, named as the 'enrichment' entry point (Figure 5.4) fulfils the needs of user story (3) where an existing openDS FDO (or reference to it) can be provided to the SDR as part of the input data.

In parallel to manipulating openDS FDOs, the Refinery gathers the minimum inputs and workflow components required to produce deterministic output and produces a Workflow-Run-RO-Crate FDO.

5.2.5 Experiments and analysis

5.2.5.1 Experimental workflows

The workflows of the SDR compose different functional components according to specific need: image segmentation, barcode reading, optical character recognition, text contextualisation/entity recognition, geocoding, taxonomic linkage, people linkage, specimen condition checking, automated identification, and data export/conversion. Broadly speaking there are two main kinds of workflow:

(i) Specimen workflows, where the specimen itself is analysed for morphological traits, colour analysis, condition checking and automated identification.

(ii) Text and label workflows, where handwritten, typed or printed text from the image is read, named entities are classified, then linked to identifiers or enhanced through post processing.

Both kinds of workflow can begin with initial openDS object creation based on the submission of specimen image files and accompanying input parameters through a forms-based user interface (*de novo* digitization entry point); or, alternatively, a pre-existing openDS object with accompanying image object(s)can be supplied as the input (enrichment entry point). Both kinds of workflow also rely on the image segmentation component as the precursor for subsequent workflow steps. Similarly, and if needed both kinds might use a format conversion and export component as their final step; for example, if an openDS FDO is not a natively compatible output for the next consuming application.

Although not within the scope of the present proof-of-concept, other more precise workflows for enhancing specific aspects of existing records can be foreseen. There are many specimen records, for example where locality text, although digitally available, is not yet geocoded. There are records with unlinked or ambiguous collector names that could be linked/disambiguated; and records where unknown specimens still need identifying.

5.2.5.2 Experimental data and evaluation

Evaluation images datasets The Refinery will be evaluated using sets of images, each composed of at least 1,000 unique specimens for each of the three categories of preserved specimen types: herbarium sheets, microscope slides and pinned insects. For herbarium sheet images we will reuse an existing benchmark dataset of 1,800 herbarium specimen images with corresponding transcribed data [Dillen 2019b]. For microscope slide and pinned insect specimen images similar evaluation datasets will be prepared against the same label characteristics: written in different languages; printed or handwritten; covering a wide range of dates; both type specimens and general collections and will provide specimens from different families and different parts of the world. Each test dataset set will be composed of images from different institutions to ensure representation of heterogeneity. For the present proof of concept, we limit the scope to Latin alphabet languages. These datasets will also be used to train Refinery models for use in tools (e.g., segmentation, named entity recognition, object/feature detection). All the datasets will be made publicly available with documentation.

Component functional tests Galaxy has a built-in functional test framework. Tools intended to become components of an SDR canonical library (actually, a Galaxy ToolShed repository) will need to pass previously defined tests within this framework. These tests, based on pre-supplied openDS FDO input and output files containing the properties expected to be populated by the tool, include validating a tool's own openDS FDO outputs by comparison against the expected output file. It will be necessary to register openDS FDOs as Galaxy custom data types.

5.2.6 Results

openDS FDOs are the core data object at the heart of the SDR, playing not only the workflow input/output role but acting also as a common data structure between tool steps within the workflow. Users can launch the workflow with either an openDS object, for further augmentation by the SDR, or they can complete a form with the specimen information, which is then converted to an openDS object before the workflow proper begins.

Each SDR Galaxy tool defines the properties it requires in JSONPath syntax [JSONPath 2023]. The wrapper validates that these properties exist in the openDS object, plucks them from the openDS JSON, makes them available as named parameters, and passes these through to the tool processing (via either a Docker or Python command line). The wrapper validates the input openDS against the openDS schema, the tool performs its processing and updates the openDS, and the wrapper validates the changed openDS against the schema before writing to disk. For the prototypical SDR, a static, local version of the openDS schema is used. Future iterations will use referenceable versions of the openDS schema, allowing for schema changes and for tools to validate their data input and outputs against versions of the schema.

On ingestion, every openDS is assigned a persistent identifier, ensuring unambiguity and referential integrity for every processed object. In production, DOIs will be minted by the DiSSCo service; for the proof-of-concept Handles with prefix `20.5000.1025/` will be used.

5.2.7 Discussion

5.2.7.1 What is being achieved?

The design of the Specimen Data Refinery uses two kinds of FAIR Digital Object—openDS FDOs and RO-Crate FDOs. Each plays a role to ensure 'FAIRer' automated digitization for natural history specimens and associated provenance capture:

- openDS FDOs act both as the input/output interface of a workflow and as the common intermediary pattern (canonical state) between steps within a workflow. They comply with DiSSCo data management principles and needs as outlined in the DiSSCo Data Management Plan [Hardisty 2019b] allowing specimen data to be processed and extended in a fully FAIR manner [Lannom 2020].
- RO-Crate FDOs record both the workflow definition and information about its configuration (shared as a method object) together with the details and context of the work done during a workflow run; details that are captured proprietarily within the adopted Galaxy environment and transformed to a common pattern (as another kind of canonical state) of provenance for later scrutiny and reproducibility of the work. These kinds of Research Objects [Bechhofer 2013] are an established mechanism whereby computational methods become first-class citizens alongside data, to be easily shared, discussed, reused and repurposed [De Roure 2010].

Both kinds of FDO are essential. They complement one another to support implementation of the FAIR principles, especially the interoperable and reusable principles by making workflows self-documenting. This renders automated whole processes (or fragments thereof) for digitizing and extending natural history specimens' data as FAIR without adding additional load to the researchers that stand to benefit most from that [Wittenburg 2022b]. Each FDO type originates from different Research Infrastructures (ELIXIR, DiSSCo) with different implementation frameworks. Yet, they interoperate effectively due to their clear roles, common conceptual model and separation of concerns.

5.2.7.2 Different FDO implementations working together

openDS FDOs have their heritage in distributed digital object services [Kahn 2006] and are implemented through Digital Object Architecture (DOA) [DONA 2021] with Digital Object Interface Protocol (DOIP) [DONA 2018], Digital Object Identifier Resolution Protocol (DO-IRP)

[Sun 2003b], and recommendations of the Research Data Alliance [Islam 2020]. Serialised as JSON, they are machine-actionable and compatible with established protocols of the World Wide Web.

RO-Crates are native to the World Wide Web, based on established web protocols, machine-readable metadata using Linked Data methods, JSON-LD and Schema.org [Bechhofer 2013, Soiland-Reyes 2022a], and community-accepted packaging mechanisms such as BagIt. This makes RO-Crates straightforward to incorporate into pre-existing platforms such as Galaxy and data repositories such as Zenodo and DataVerse.

Both kinds of FDO use Persistent identifiers (PID), allowing instances to be both uniquely identified and their location to be determined; RO-Crates, as web natives, use URIs whereas openDS, as DOA objects, use Handle PIDs. Instances of both kinds are described by metadata and contain or reference data.

RO-Crates are self-describing using a metadata file and use openly-extensible profiles to type the Crates (profile-typing) to set out expectations for their metadata and content. openDS uses an object-oriented object typing and instance approach to define the structure and content of data/metadata. Complex object types are constructed from basic types, an extension-section basic type. Both approaches seek to avoid locking objects into repository silos, ensuring that FDO instances can be interpreted outside of the contexts in which they were originally created/stored.

Structurally and semantically openDS FDOs and RO-Crate FDOs are potentially isomorphic, although at different granularity levels. Their main difference is in method calling. As a DOA object, openDS would expect to respond to type-specific method calls if these were implemented. RO-Crates delegate actionability to applications that interpret their self-describing profile.

Within the SDR the two kinds of FDO fulfill distinct and interlocking roles for data (openDS) and self-documented method (RO-Crate) so their different forms is not an issue. In future there may be a need to map and convert between the approaches (e.g., for reconstructing past processing), which would be assisted by the common FDO conceptual model [Bonino 2019].

5.2.7.3 Key domain challenges ahead

For a digitized specimen to conform to FAIR principles, its data must be linked to a vocabulary of terms, but choosing a single vocabulary is likely to cause interoperability issues when cross-linking to resources using another vocabulary, for example Darwin Core, Schema.org, or Access to Biological Collection Data (ABCD). Whilst concepts can be mapped across vocabularies (for example, using Simple Knowledge Organization System (SKOS) matching), such an effort might rapidly become overly complex and cumbersome, as the challenge of the Unified Medical Language System (UMLS) demonstrates. The challenge remains - how is such a mapping exercise maintained at a 'just enough' level?

Different Earth Science domains have different use cases for digital records. A digital record produced for biodiversity research is likely to have different granularity, understanding and focus to one produced for climate science. It remains to be seen if a single FAIR Digital Object definition could be produced to satisfy multiple domains, and if different objects could be produced for different domains, what would they look like; and would this hinder future cross-compatibility?

The openDS FDO type produced by the SDR is a new object format for the natural history domain that is foreseen to become an adopted standard over time. Institutional collection management systems will need to be upgraded before they can consume the FDO outputs from the SDR. Early adopters may need assistance to produce SDR exporters matching proprietary ingestion formats. For an interim period, there may be a need for the SDR to output today's widely used Darwin Core Archives format in parallel.

As the functional requirements of the SDR are emergent, a minimum viable product has been scoped, but this should be contrasted with the notion of a useful product. An MVP is a prototype; a tool to get a project off the ground with enough features to be usable by early adopters, and to build on to learn user requirements. But it is not intended to meet the day-to-day requirements of all users. To nurture future development, care must be taken to continue involving key stakeholders in eliciting further requirements to make the SDR useful for the widest range of users, and from there, develop a rich, configurable tool to allow simple uptake and provide utility for resource-poor collections.

5.2.8 Conclusion and Future Work

The Specimen Data Refinery is likely to garner widespread interest across the Natural History community. Whilst the promise of a scalable, community-driven digitization platform is tantalising for many natural history professionals, the Specimen Data Refinery project is still in its early stages, and, as discussed above key challenges lie ahead.

Although natural history collections are generally catalogued by the taxonomic identity of the curated object, there remains a large historical backlog of unidentified specimens. The Meise Botanic Garden (BE), for example, has an estimated 4 million specimens with at least 11% not yet identified to species level. Furthermore, it is calculated that half of the World's estimated 70,000 plant species yet to be described have already been collected and are waiting in collections still to be 'discovered' [Crusoe 2022]. The same is likely to be true for other groups of organisms, especially insects. Unnamed specimens tend to have lowest priority for digitization and their data are rarely shared. Machine learning as canonical steps in SDR workflows presents a tremendous opportunity to put an identification on these specimens and potentially, to triage them for further taxonomic investigation.

5.3 Incrementally building FAIR Digital Objects with Specimen Data Refinery workflows

Specimen Data Refinery (SDR) is a developing platform for automating transcription of specimens from natural history collections [Hardisty 2022] (Section 5.2 on page 133). SDR is based on computational workflows and digital twins using FAIR Digital Objects.

We show our recent experiences with building SDR using the Galaxy workflow system and combining two FDO methodologies with open digital specimens (openDS) and RO-Crate data packaging. We suggest FDO improvements for incremental building of digital objects in computational workflows.

5.3.1 SDR workflows

SDR[66] is realised as the workflow system Galaxy [Afgan 2018] with SDR tools[67] installed. An Open Research challenge is that some tools have machine learning models with a commercial licence. This complicates publishing to Galaxy toolshed[68]—however, we created Ansible[69] scripts to install equivalent Galaxy servers, including tools and dependencies, accounts and workflows. SDR workflows are published in WorkflowHub[70] as FDOs.

We implemented the use case *De novo digitization* in Galaxy [Brack 2022b]. Shown in Figure 5.5 on the facing page the workflow steps exchange openDS JSON [Hardisty 2019a], for incremental completion of a digital specimen. Initial stages build a template openDS from a CSV with metadata and image references—subsequent analysis completes the rest of the JSON with *regions* of interest, *text* digitised from handwriting, and recognised *named entities*.

Galaxy can visualise outputs of each step (Figure 5.6 on the next page), important to make the FDOs understandable by domain experts and to verify accuracy of SDR.

We are adding workflows for partial stages, e.g. detection of regions [Livermore 2022a] and hand-written text recognition [Livermore 2022b], which we will combine with scalability testing and wider testing by project users. Additional workflows will enhance existing FDOs and use new tools such as barcode detection of museums' internal identifiers.

We are now ready to publish digital specimens as FAIR Digital Objects, with registration into DiSSCO repositories[71], PID assignment and workflow provenance. However, even at this early stage we have identified several challenges that need to be addressed.

[66] https://sdr.nhm.ac.uk/
[67] https://github.com/DiSSCo/SDR
[68] https://toolshed.g2.bx.psu.edu/
[69] https://www.ansible.com/
[70] https://workflowhub.eu/projects/72
[71] https://www.dissco.eu/dissco/technical-infrastructure/

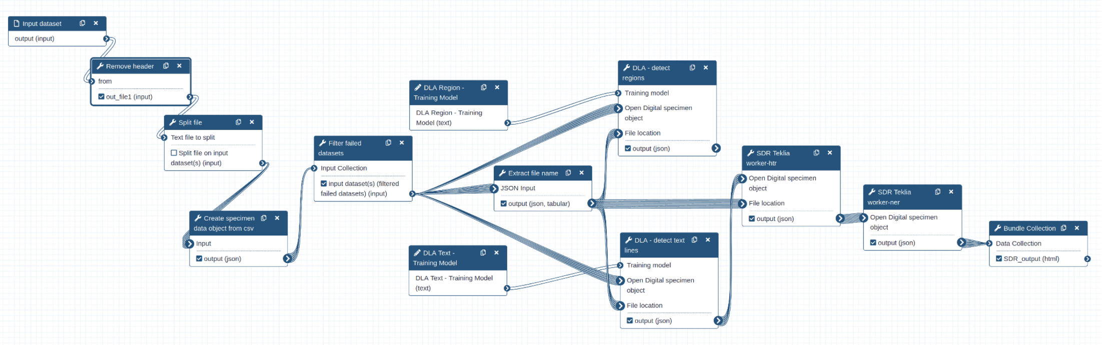


Figure 5.5: FDO propagation in workflow. Draft Galaxy workflow [Brack 2022b] shows propagation of partial Open Digital Specimen FDOs between individual canonical workflow building blocks. First steps process a CSV file to create the initial openDS, where referenced images are analysed to detect text lines which are OCRed and then recognised as named entities. Bands indicate flow of collections of openDS, processed concurrently by each step. The final step bundles the collection of openDS FDOs as JSON files in a ZIP archive.

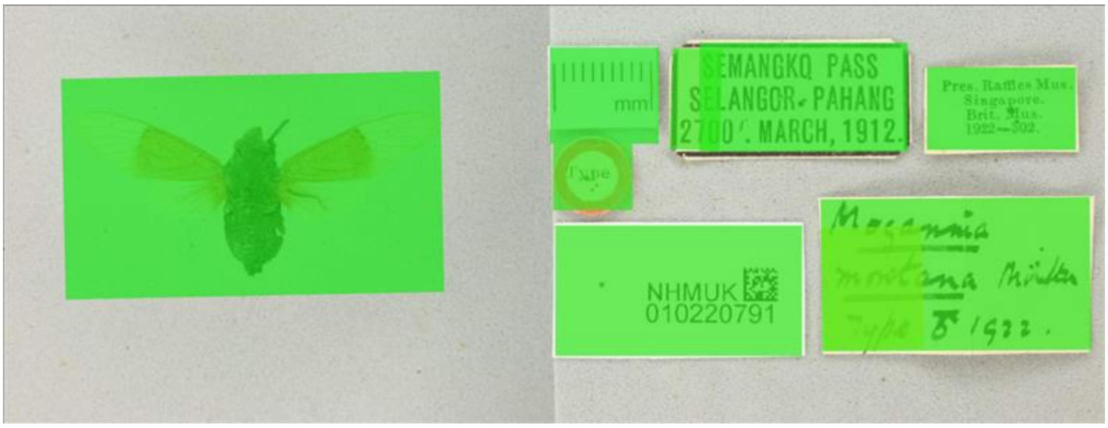


Figure 5.6: Visualising openDS FDO within Galaxy. Showing detected regions of interest (specimen, labels and scale bar) for a pinned insect.

5.3.2 FDO lessons

We highlight the *de novo* use case because this workflow is exchanging *partial* FDOs—openDS objects which are not fully completed and not yet assigned persistent identifiers. openDS schemas[72] are still in development, therefore SDR uses a more flexible JSON schema[73] where only the initial metadata (populated from CSV) are required. Each step validates the partial FDO before passing it to the underlying command line tool.

Although workflow steps exchange openDS objects, they cannot be combined in any order. For instance, *named entity recognition* requires digitised text in the FDO. We can consider these intermediate steps as *sub-profiles* of an FDO Type. Unlike hierarchical subclasses, these FDO profiles are more like ducktyping[74]. For instance a *text detection* step may only require the `regions` key, but semantically there is no requirement for say `OpenDSWithText` to be a subclass of `OpenDSWithRegion`, as text also can be transcribed manually without regions.

Similarly, we found that some steps can be executed in parallel, but this requires merging of partial FDOs. This can be achieved by combining JSON queries and JSON Schemas, but indicates that it may be more beneficial to have FDO fragments as separate objects. Adding openDS fragment steps would, however, complicate workflows.

Several of our tools process the referenced images, currently `https` URLs in openDS. We added a caching layer to avoid repeated image downloading, coupled with local file-paths wiring in the workflow. A similar challenge occurs if accessing image data using DOIP, which unlike HTTP, has no caching mechanisms.

5.3.3 RO-Crate lessons

Galaxy is developing[75] support for importing and exporting Workflow Run Crates[76], a profile of RO-Crate [Soiland-Reyes 2022a] to captures execution history of a workflow, including its definition and intermediate data [De Geest 2022]. SDR is adopting this support to combine openDS FDOs with workflow provenance, as envisioned by [Walton 2020a].

Our prototype *de novo* workflow returns results as a ZIP file of openDS objects. End-users should also get copies of the referenced images and generated visualisations, along with workflow execution metadata. We are investigating ways to embed the preliminary Galaxy workflow history before the final step, so that this result can be an enriched RO-Crate.

5.3.4 Conclusions

SDR is an example of machine-assisted construction of FDOs, which highlight the needs for intermediate digital objects that are not yet FDO compliant. The passing of such “local FDOs” is

[72] https://github.com/DiSSCo/openDS

[73] https://github.com/DiSSCo/SDR/blob/main/galaxy-workflow/config/opends-schema.json

[74] https://en.wikipedia.org/wiki/Duck_typing

[75] Completed after publication of this article, see Section 5.4.3.2 on page 165.

[76] https://www.researchobject.org/workflow-run-crate/ see also Section 5.4 on page 152.

beneficial not just for efficiency and visual inspection, but also to simplify workflow composition of canonical workflow building blocks. At the same time we see that it is insufficient to only pass FDOs as JSON objects, as they also have references to other data such as images, which should not need to be re-downloaded.

Further work will investigate the use of RO-Crate as a wrapper of partial FDOs, but this needs to be coupled with more flexible FDO types as profiles, in order to restrict "impossible" ordering of steps depending on particular inner FDO fragments. A distinction needs to be made between open digital specimens that are in "draft" state and those that can be pushed to DiSSCo registries.

We are experimenting with changing the SDR components into Canonical Workflow Building Blocks [Soiland-Reyes 2022b] (Section 5.1 on page 121) using the Common Workflow Language [Crusoe 2022]. This gives flexibility to scalably execute SDR workflows on different compute backends such as HPC or local cluster, without the additional setup of Galaxy servers.

5.4 Recording provenance of workflow runs with RO-Crate

Recording the provenance of scientific computation results is key to the support of traceability, reproducibility and quality assessment of data products. Several data models have been explored to address this need, providing representations of workflow plans and their executions as well as means of packaging the resulting information for archiving and sharing. However, existing approaches tend to lack interoperable adoption across workflow management systems.

In this work we present **Workflow Run RO-Crate**, an extension of Research Object Crate (RO-Crate) and Schema.org to capture the provenance of the execution of computational workflows at different levels of granularity and bundle together all their associated products (inputs, outputs, code, etc.). The model is supported by a diverse, open community that runs regular meetings, discussing development, maintenance and adoption aspects. Workflow Run RO-Crate is already implemented by several workflow management systems, allowing interoperable comparisons between workflow runs from heterogeneous systems. We describe the model, its alignment to standards such as W3C PROV, and its implementation in six workflow systems. Finally, we illustrate the application of Workflow Run RO-Crate in two use cases of machine learning in the digital image analysis domain.

5.4.1 Introduction

A crucial part of scientific research is recording the provenance of its outputs. The W3C PROV standard defines provenance as "a record that describes the people, institutions, entities, and activities involved in producing, influencing, or delivering a piece of data or a thing" [Moreau 2013]. Provenance is instrumental to activities such as traceability, reproducibility, accountability, and quality assessment [Herschel 2017]. The constantly growing size and complexity of scientific datasets and the analysis that is required to extract useful information from them has made science increasingly dependent on advanced automated processing techniques in order to get from experimental data to final results [Himanen 2019, Gauthier 2019, Huntingford 2019]. Consequently, a large part of the provenance information for scientific outputs consists of descriptions of complex computer-aided data processing steps. This data processing is often expressed as workflows, i.e., high-level applications that coordinate multiple tools and manage intermediate outputs in order to produce the final results.

In order to homogenise the collection and interchange of provenance records, the W3C consortium proposed the PROV-O standard [Lebo 2013a], an OWL [W3C 2012] representation of PROV for provenance in the Web. PROV-O has been widely extended for workflows (D-PROV [Missier 2013], ProvONE [Cuevas-Vicenttín 2016], OPMW [Garijo 2011], P-PLAN [Garijo 2012]), where provenance information is collected in two main forms: prospective and retrospective [Freire 2008]. *Prospective provenance*—the execution plan—is essentially the workflow itself: it includes a machine-readable specification with the processing steps to be performed and the data and software dependencies to carry out each computation. *Retrospective provenance* refers to what actually happened during an execution, i.e. what were the values of the input parameters, which outputs were produced, which tools were executed, how much time did the execution take, whether the execution was successful or not, etc. Retrospective provenance can also be represented at different levels of abstraction depending on available computing resources: for instance, by the workflow execution becoming a single activity which produces results, by specifying the individual execution of each workflow step, or by going a step further and indicating how each step is divided into sub-processes when a workflow is deployed in a cluster.

Different workflow systems have adopted and extended PROV (and its PROV-O representation) to the workflow domain (WINGS [Gil 2011, Garijo 2014b], VisTrails [Scheidegger 2008, Costa 2013]), in order to ease the burden of provenance collection from tool developers to Workflow Management Systems (WfMSs) [Atkinson 2017, Pérez 2018].

D-PROV, PROV-ONE, OPMW-PROV, P-Plan propose representations of workflow plans and their respective executions, taking into account the features of the workflow systems implementing them (e.g., hierarchical representations, sub-processes, etc.). Other data models like *wfprov* and *wfdesc* [Belhajjame 2015] go a step further by considering not only the link between plans and executions, but how to package the various artefacts as a Research Object [Bechhofer 2013] in order to ease portability while keeping the context of a digital experiment.

However, while these models address some workflow provenance representation issues, they have two main limitations: Firstly, the extensions of PROV are not directly interoperable because of differences in granularity or different assumptions in their workflow representations; secondly, their support from WfMS is typically one system per model. An early approach to unify and integrate workflow provenance traces across WfMS was WEST (Workflow Ecosystems through STandards) [Garijo 2014b], through the use of WINGS [Gil 2011] to build workflow templates and different converters.

In all of these workflow provenance models, the emphasis is on the workflow execution structure as a directed graph, with only partial references for the data items. The REPRODUCE-ME ontology [Samuel 2022] extended PROV and P-Plan to explain the overall scientific process with the experimental context including real life objects (e.g. instruments, specimens) and human activities (e.g. lab protocols, screening), demonstrating provenance of individual Jupyter Notebook cells[77] and highlighting the need for provenance also where there is no workflow management system.

More recently, interoperability have been partially addressed by Common Workflow Language Prov (CWLProv) [Khan 2019], which represents workflow enactments as ROs serialised according to the Big Data Bag (BDBag) approach [Chard 2016]. The resulting format is a folder containing several data and metadata files [Soiland-Reyes 2018], expanding on the RO Bundle approach of Taverna [Soiland-Reyes 2016]. CWLProv also extends PROV with a representation of executed processes (activities), their inputs and outputs (entities) and their executors (agents), together with their Common Workflow Language (CWL) specification [Crusoe 2022]—a stand-

[77] https://sheeba-samuel.github.io/REPRODUCE-ME/research/provbook.html

ard workflow specification adopted by at least a dozen different workflow systems[78]. Although CWLProv includes prospective provenance as a *plan* within PROV (based on the *wfdesc* model), in practice its implementation does not include tool definitions or file formats, as proposed by the wfdesc extension Roterms[79]. In order for CWLProv consumers to reconstruct the full prospective provenance for understanding the workflow, they would also need to inspect the separate workflow definition in the native language of the WfMS. Additionally, the CWLProv RO may include several other metadata files and PROV serialisations conforming to different formats, complicating its generation and consumption.

As for granularity, CWLProv proposed multiple levels of provenance [Khan 2019, figure 2], from Level 0 (capturing workflow definition) to Level 3 (domain-specific annotations). In practice, the CWL reference implementation *cwltool* [Amstutz 2023] and the corresponding CWLProv specification [Soiland-Reyes 2018] records provenance details of all task executions together with the intermediate data and any nested workflows (CWLProv level 2), a granularity level that requires substantial support from the WfMS. This approach is appropriate for workflow languages where the execution plan, including its distribution among the various tasks, is known well in advance (such as CWL). However, it can be at odds with other systems where the execution is more dynamic, depending on the verification of specific runtime conditions, such as the size and distribution of the data (e.g., COMPSs [Lordan 2014]).

This makes the implementation of CWLProv challenging, as shown by the fact that at the time of writing the format is supported only by cwltool. Finally, being based on the PROV model, CWLProv is highly focused on the interaction between agents, processes and related entities, while support for contextual metadata (such as workflow authors, licence or creation date) in the Research Object Bundle is limited[80] and stored in a separate manifest file, that includes the data identifier mapping to filenames. A project that uses serialised ROs similar to those used by CWLProv is Whole Tale [Chard 2019], a web platform with a focus on the narrative around scientific studies and their reproducibility, where the serialised ROs are used to export data and metadata from the platform. In contrast, our work is primarily focused on the ability to capture the provenance of computational workflow execution including its data and executable workflow definitions.

RO-Crate [Soiland-Reyes 2022a] is a recent approach to packaging research data together with their metadata; it extends Schema.org [Guha 2015], a popular vocabulary for describing resources on the Web. In its simplest form, an RO-Crate is a directory structure that contains a single JSON-LD [Sporny 2020] metadata file at the top level. The metadata file describes all entities stored in the RO-Crate along with their relationships; it is both machine-readable and human-readable. RO-Crate is general enough to be able to describe any dataset, but can also be made as specific as needed through the use of extensions called *profiles*. At the same time, the broad set of types and properties from Schema.org, complemented by a few additional

[78] https://www.commonwl.org/implementations/
[79] https://wf4ever.github.io/ro/2016-01-28/roterms
[80] https://w3id.org/bundle/context

terms from other vocabularies, make the RO-Crate model capable of expressing a wide range of contextual information that complements and enriches the core information specified by the profile. This may include, among others, the workflow authors and their affiliations, associated publications, licensing information, related software, etc. This is an approach used by WorkflowHub [Goble 2021], a workflow system agnostic workflow registry which specifies a Workflow RO-Crate profile [Bacall 2022] to gather the workflow definition with such metadata in an archived RO-Crate [81].

In this work, we present **Workflow Run RO-Crate** (WRROC), an extension of RO-Crate for representing computational workflow execution provenance. Our main contributions are the following:

- A collection of RO-Crate profiles to represent and package both the prospective and the retrospective provenance of a computational workflow run in a way that is machine-actionable [Batista 2022], independent of the specific workflow language or execution system, and including support for re-execution.
- Implementations of the model in six workflow management systems and one conversion tool.
- A mapping of our profiles against the W3C PROV-O Standard using the Simple Knowledge Organisation System (SKOS) [Isaac 2009].

To foster usability, the profiles are characterised by different levels of detail, and the set of mandatory metadata items is kept to a minimum in order to ease the implementation. This flexible approach increases the model's adaptability to the diverse landscape of WfMS used in practice. The base profile, in particular, is applicable to any kind of computational process, not necessarily described in a formal workflow language. All profiles are supported and sustained by the Workflow Run RO-Crate community, which meets regularly to discuss extensions, issues and new implementations.

The rest of this section is organised as follows: we first describe the Workflow Run RO-Crate profiles; we then illustrate implementations and usage examples; this is followed by a discussion and plans for future work.

5.4.2 The Workflow Run RO-Crate profiles

RO-Crate profiles are extensions of the base RO-Crate specification that describe how to represent the entities and relationships that appear in a specific domain or use case[82]. An RO-Crate conforming to a profile is not just machine-readable, but also machine-actionable as a digital object whose type is represented by the profile itself [Soiland-Reyes 2022c].

The Workflow Run RO-Crate profiles are the main outcome of the activities of the Workflow Run

[81] See Section 4.1.4.1 on page 94.

[82] See Sections 4.1.4 on page 93 and 6.1.2.4 on page 191.

RO-Crate Community[83], an open working group that includes workflow users and developers, WfMS users and developers, and researchers and software engineers interested in workflow execution provenance and Findable, Accessible, Interoperable and Reusable (FAIR) approaches for data and software. In order to develop the Workflow-Run RO-Crate profiles, one of the first community efforts was to compile a list of requirements in the form of competency questions[84] to be addressed by the model. Each requirement was backed up by a rationale and linked to a GitHub issue to drive the public discussion forward. When a requirement was addressed, related changes were integrated into the profiles and the relevant issue was closed. Many of the original issues are now closed, and the profiles have had four official releases on Zenodo.

As requirements were being defined, it became apparent that one single profile would not have been sufficient to cater for all possible usage scenarios. In particular, while some use cases required a detailed description of all computations orchestrated by the workflow, others were only concerned with a "black box" representation of the workflow and its execution as a whole (i.e., whether the execution was successful and which results were obtained). Additionally, some computations involve a data flow across multiple applications that are executed without the aid of a WfMS and thus are not formally described in a standard workflow language. These observations led to the development of three profiles:

(1) Process Run Crate[85] to describe the execution of one or more tools that contribute to a computation.

(2) Workflow Run Crate[86] to describe a computation orchestrated by a predefined workflow.

(3) Provenance Run Crate[87] to describe a workflow computation including the internal details of individual step executions.

In the rest of this section we describe each of the above profiles in detail. We use italics to denote the types and properties describing entities and their relationships: these are defined in the RO-Crate JSON-LD context[88], which extends Schema.org with terms from the Bioschemas [Gray 2017] ComputationalWorkflow profile[89] and other vocabularies. More specifically, from Bioschemas we use the *ComputationalWorkflow* and *FormalParameter* types as well as the *input* and *output* properties. Note that these terms, though coming from Bioschemas, are not specific to the life sciences. We also developed a context extension through a dedicated "workflow-run" namespace[90] to represent concepts that are not captured by terms in the RO-Crate context.

[83] https://www.researchobject.org/workflow-run-crate

[84] https://www.researchobject.org/workflow-run-crate/requirements

[85] https://w3id.org/ro/wfrun/process

[86] https://w3id.org/ro/wfrun/workflow

[87] https://w3id.org/ro/wfrun/provenance

[88] https://www.researchobject.org/ro-crate/1.1/context.jsonld

[89] https://bioschemas.org/profiles/ComputationalWorkflow/1.0-RELEASE

[90] https://w3id.org/ro/terms/workflow-run#

5.4.2.1 Process Run Crate

The Process Run Crate profile [WRROC 2023a] contains specifications on describing the execution of one or more software applications that contribute to the same overall computation, but are not necessarily coordinated by a top-level workflow or script. For instance, they could be executed manually by a human agent, one after the other as intermediate datasets become available, as shown in the process run crate[91] from [Meurisse 2023].

Being the basis for all profiles in the WRROC collection, Process Run Crate specifies how to describe the fundamental entities involved in a computational run: a software application (represented by a *SoftwareApplication*, *SoftwareSourceCode* or *ComputationalWorkflow* entity) and its execution (represented by a *CreateAction* entity), with the latter linking to the former via the *instrument* property. Other important properties of the *CreateAction* entity are *object*, which links to the action's inputs, and *result*, which links to its outputs. The time the execution started and ended can be provided, respectively, via the *startTime* and *endTime* properties. The *Person* or *Organization* entity that performed the action is referred to via the *agent* property. Figure 5.7 on the following page shows the entities used in Process Run Crate together with their relationships.

As an example, suppose a user called John Doe runs the `head` UNIX command to extract the first ten lines of an input file named `lines.txt`, storing the result in another file called `selection.txt`. John then runs the `sort` command on `selection.txt`, storing the sorted output in a new file named `sorted_selection.txt`. Figure 5.8 on page 159 contains a diagram of the two actions and their relationships to the other entities involved. Note how the actions are connected by the fact that the output of "Run Head" is also the input of "Run Sort": they form an "implicit workflow", whose steps have been executed manually rather than by a software tool.

Process Run Crate extends the RO-Crate guidelines on representing software used to create files with additional requirements and conventions. This arrangement is typical of the RO-Crate approach, where the base specification provides general recommendations to allow for high flexibility, while profiles—being more concerned with the representation of specific domains and machine actionability—provide more detailed and structured definitions. Nevertheless, in order to be broadly applicable, profiles also need to avoid the specification of too many strict requirements, trying to strike a good trade-off between flexibility and actionability. One of the implications of this approach is that consumers need to code defensively, avoiding unwarranted assumptions—e.g. by verifying that a value exists for an optional property before trying to retrieve it and use it.

5.4.2.2 Workflow Run Crate

The Workflow Run Crate profile [WRROC 2023b] combines the Process Run Crate and WorkflowHub's Workflow RO-Crate [Bacall 2022] profiles to describe the execution of "proper" computational workflows managed by a WfMS. Such workflows are typically written in a special-purpose language, such as CWL or Snakemake [Köster 2012], and run by one or more

[91] https://w3id.org/ro/doi/10.5281/zenodo.6913045

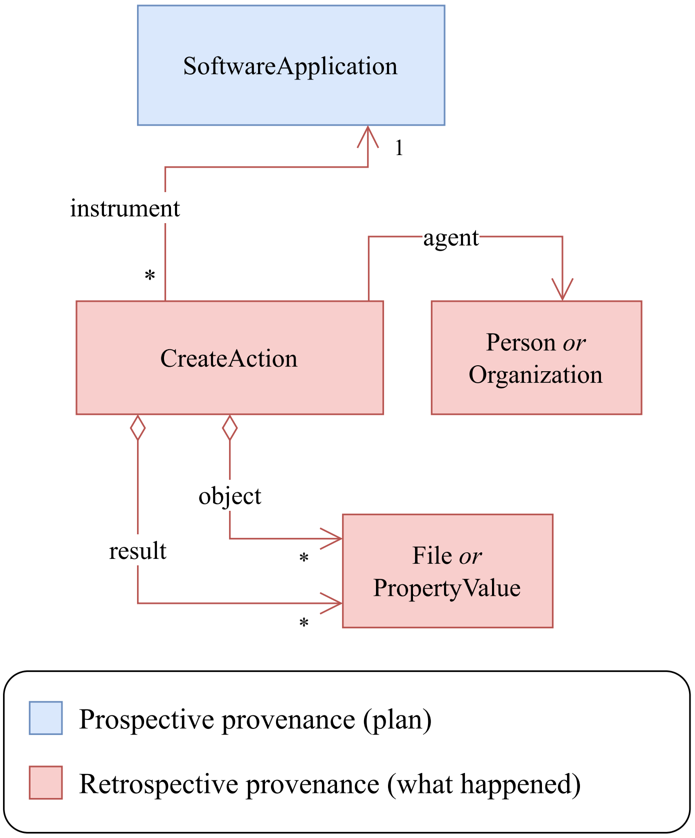


Figure 5.7: UML class diagram for Process Run Crate. The central entity is the *CreateAction*, which represents the execution of an application. It relates with the application itself via *instrument*, with the entity that executed it via *agent* and with its inputs and outputs via *object* and *result*, respectively. *File* is an RO-Crate alias for Schema.org's *MediaObject*. Some inputs (and, less commonly, outputs), however, are not stored as files or directories, but passed to the application (e.g., via a command line interface) as values of various types (e.g., a number or string). In this case, the profile recommends a representation via *PropertyValue*. For simplicity, we left out the rest of the RO-Crate structure (e.g. the root *Dataset*). In this UML class notation diamond $\Diamond$ arrows indicate aggregation and regular arrows indicate references, $*$ indicates multiple instances, 1 means single instance.

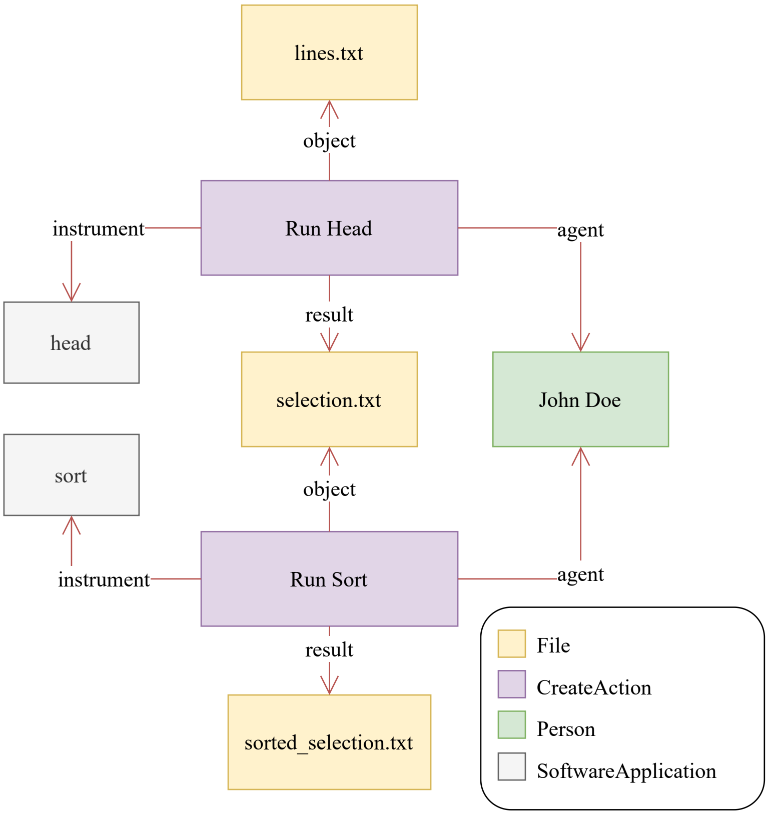


Figure 5.8: Diagram of a simple workflow where the `head` and `sort` programs were run manually by a user. The executions of the individual software programs are connected by the fact that the file output by `head` was used as input for `sort`, documenting the computational flow in an implicit way. Such executions can be represented with Process Run Crate.

WfMS (e.g., StreamFlow [Colonnelli 2021], Galaxy [Galaxy 2022]). As in Process Run Crate, the execution is described by a *CreateAction* that links to the application via *instrument*, but in this case the application must be a workflow, as prescribed by Workflow RO-Crate. More specifically, Workflow RO-Crate states that the RO-Crate must contain a main workflow typed as *File*, *SoftwareSourceCode* and *ComputationalWorkflow*. The execution of the individual workflow steps, instead, is not represented: that is left to the more detailed Provenance Run Crate profile (described in the next Section 5.4.2.3 on the facing page).

The Workflow Run RO-Crate profile also contains recommendations on how to represent the workflow's input and output parameters, based on the aforementioned Bioschemas [Gray 2017] ComputationalWorkflow profile. All these elements are represented via the *FormalParameter* entity and are referenced from the main workflow via the *input* and *output* properties. While the entities referenced from *object* and *result* in the *CreateAction* represent data entities and argument values that were actually used in the workflow execution, the ones referenced from *input* and *output* correspond to formal parameters, which acquire a value when the workflow is run (see Figure 5.9 on the next page). In the profile, the relationship between an actual value and the corresponding formal parameter is expressed through the *exampleOfWork* property—the downloadable file is a realisation of the formal parameter definition. For instance, in the JSON-LD snippet of Listing 5.3 a formal parameter (`#annotations`) is illustrated together with a corresponding `final-annotations.tsv` file:

```
{
    "@id": "#annotations",
    "@type": "FormalParameter",
    "additionalType": "File",
    "encodingFormat": "text/tab-separated-values",
    "valueRequired": "True",
    "name": "annotations"
},
{
    "@id": "final-annotations.tsv",
    "@type": "File",
    "contentSize": "14784",
    "exampleOfWork": {"@id": "#annotations"}
}
```

Listing 5.3: Relating an actual value to its formal parameter definition. The Bioschemas *FormalParameter*[92] entity `#annotations` defines possible values for a workflow paramater named `annotations`. `#final-annotations.tsv`, a downloadable *File*, is an `exampleOfWork` in the sense that it realises the parameter definition. It is also possible to flag particular values as representative exemplar values with the reverse `workExample` property from the *FormalParameter*, which is not the case here.

Figure 5.9 on the next page shows the entities used in Workflow Run Crate together with their

[92] https://bioschemas.org/profiles/FormalParameter/1.0-RELEASE

relationships.

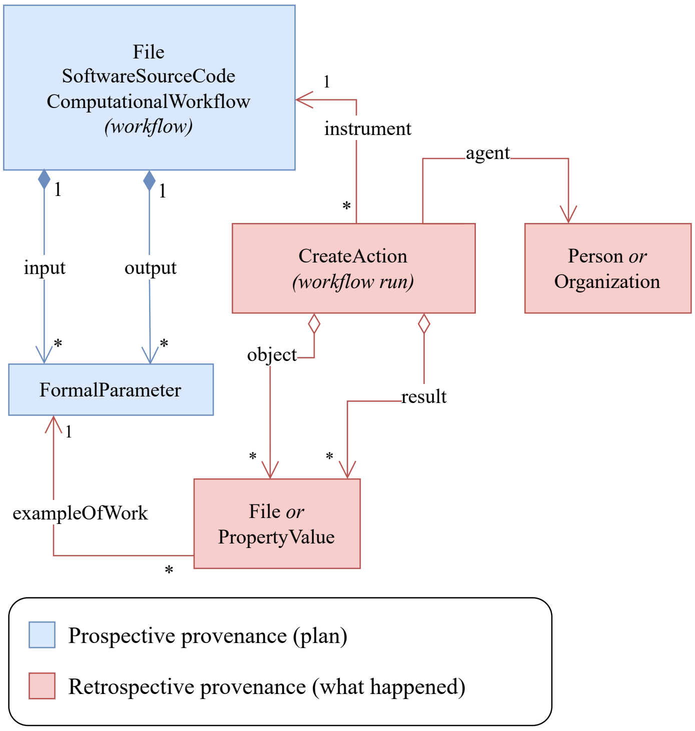


Figure 5.9: UML class diagram for Workflow Run Crate. The main differences with Process Run Crate are the representation of formal parameters and the fact that the application is expected to be an entity with types *File*, *SoftwareSourceCode* and *ComputationalWorkflow*. Effectively, the entity belongs to all three types, and its properties are the union of the properties of the individual types. The filled diamond ♦ indicates composition, empty diamond ◊ aggregation, and other arrows relations.

5.4.2.3 Provenance Run Crate

The Provenance Run Crate profile [WRROC 2023c] extends Workflow Run Crate by adding new concepts to describe the internal details of a workflow run, including individual tool executions, intermediate outputs and related parameters. Individual tool executions are represented by additional *CreateAction* instances that refer to the tool itself via *instrument*—analogously to its use in Process Run Crate. The workflow is required to refer to the tools it orchestrates through the *hasPart* property, as suggested in the Bioschemas ComputationalWorkflow profile, though in the latter it is only a recommendation.

To represent the logical steps defined by the workflow, this profile uses *HowToStep*[93] i.e., "A step in the instructions for how to achieve a result". Steps point to the corresponding tools via the *workExample* property and are referenced from the workflow via the *step* property; the execution of a step is represented by a *ControlAction* pointing to the *HowToStep* via *instrument* and to the

[93] https://schema.org/HowToStep

CreateAction instance(s) that represent the corresponding tool execution(s) via *object*. Note that a step execution does not coincide with a tool execution: an example where this distinction is apparent is when a step maps to multiple executions of the same tool over a list of inputs (e.g. the "scattering" feature in CWL).

An RO-Crate following this profile can also represent the execution of the WfMS itself (e.g., cwltool) via *OrganizeAction*, pointing to a representation of the WfMS via *instrument*, to the steps via *object* and to the workflow run via *result*. The *object* attribute of the *OrganizeAction* can additionally point to a configuration file containing a description of the settings that affected the behaviour of the WfMS during the execution.

Figure 5.10 on the facing page shows the various entities involved in the representation of a workflow run via Provenance Run Crate together with their relationships.

This profile also includes specifications on how to describe connections between parameters. Parameter connections—a fundamental feature of computational workflows—describe (i) how tools take as input the intermediate outputs generated by other tools and (ii) how workflow-level parameters are mapped to tool-level parameters. For instance, consider again the workflow depicted in Figure 5.8 on page 159, and suppose it is implemented in a workflow language such as CWL. The workflow-level input (a text file) is connected to the input of the "head" tool wrapper, and the output of the latter is connected to the input of the "sort" tool wrapper.

A representation of parameter connections is particularly useful for traceability, since it allows to document the inputs and tools on which workflow outputs depend. Since the current RO-Crate context has no suitable terms for the description of such relationships, we added appropriate ones to the aforementioned "workflow-run" context extension (the https://w3id.org/ro/terms/workflow-run# namespace): a *ParameterConnection* type with *sourceParameter* and *targetParameter* attributes that respectively map to the source and target formal parameters, and a *connection* property to link from the relevant step or workflow to the *ParameterConnection* instances.

This profile is the most detailed of the three, and offers the highest level of granularity. Fig. 5.11 on page 164 shows the relationship between the specifications of the profiles as a Venn diagram.

5.4.3 Implementations

Support for the Workflow Run RO-Crate profiles presented in this work has been implemented in a number of systems, showing support from the community and demonstrating their usability in practice. We describe seven of these implementations (one in a conversion tool and six in WfMS) in the following sections. These tools have been developed in parallel by different teams, and independently from each other. RO-Crate has a strong ecosystem of tools [Soiland-Reyes 2022a] (Section 4.1.3 on page 91), and the WRROC implementations have either re-used these or added their own approach to the standards.

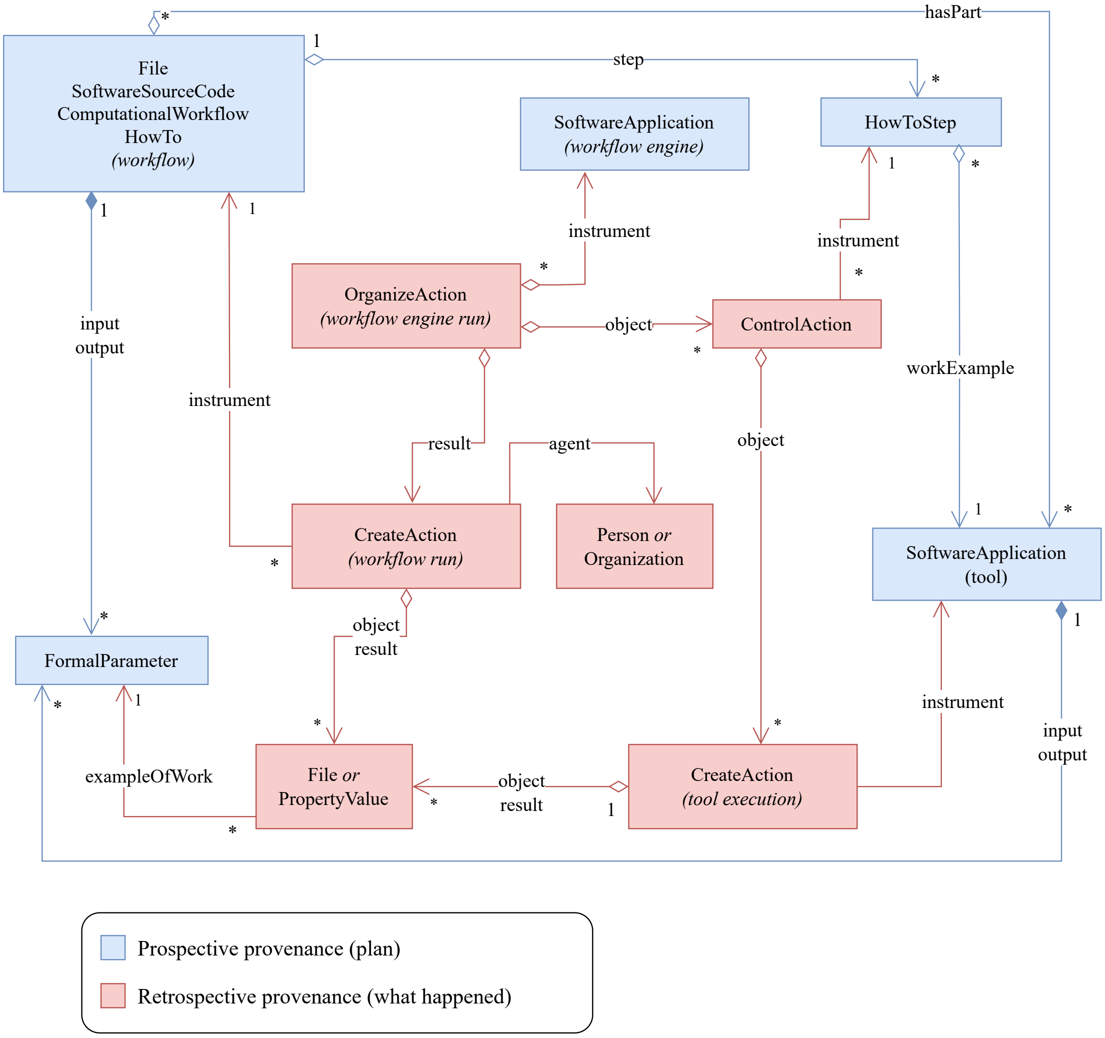


Figure 5.10: UML class diagram for Provenance Run Crate. In addition to the workflow run, this profile represents the execution of individual steps and their related tools.

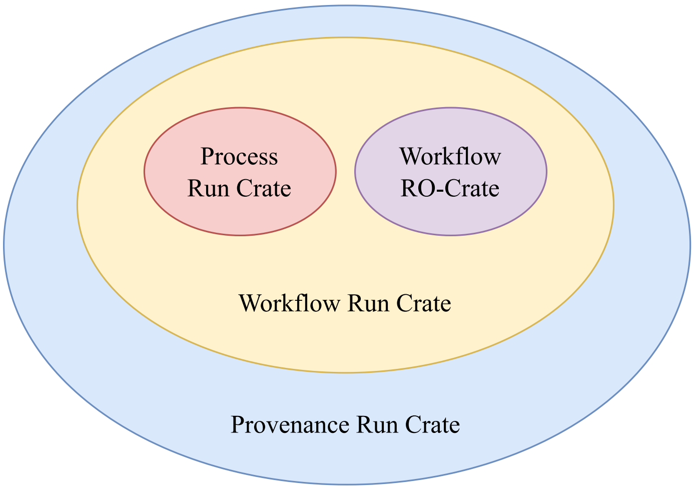


Figure 5.11: Venn diagram of the specifications for the various RO-Crate profiles. Workflow Run Crate inherits the specifications of both Process Run Crate and Workflow RO-Crate. Provenance Run Crate, in turn, inherits the specifications of Workflow Run Crate.

5.4.3.1 Runcrate

Runcrate[94] [Leo 2023a] is a Workflow Run RO-Crate toolkit which also serves as a reference implementation of the proposed profiles. It consists of a Python package with a command line interface, providing a straightforward path to integration in Python software and other workflows. The runcrate toolkit includes functionality to convert CWLProv ROs to RO-Crates conforming to the Provenance Run Crate profile (*runcrate convert*), effectively providing an indirect implementation of the format for cwltool. Indeed, the CWLProv model provided a basis for the Provenance Run Crate profile, and the implementation of a conversion tool in runcrate at times drove the improvement and extension of the profile as new requirements or gaps in the old designs emerged. Runcrate converts both the retrospective provenance part of the CWLProv RO (the RDF graph of the workflow's execution) and the prospective provenance part (the CWL files, including the workflow itself). Both parts are thus converted into a single, workflow language-agnostic metadata resource.

Another functionality offered by the runcrate package is *runcrate report* which reports on the various executions described in an input RO-Crate, listing their starting and ending times, the values of the various parameters, etc. (example output in Listing 5.4 on page 173). Runcrate report demonstrates how the provenance profiles presented in this work enable comparison of runs interoperably across different workflow languages or different implementations of the same language. This functionality has also been used as a lightweight validator for the various implementations.

We also added a *run* subcommand to re-execute the computation described by an input Workflow

[94] https://github.com/ResearchObject/runcrate

Run Crate or Provenance Run Crate where CWL was used as a workflow language. It works by mapping the RO-Crate description of input parameters and their values (the workflow's *input* and the action's *object*) to the format expected by CWL, which is then used to relaunch the workflow on the input data. This functionality shows the machine-actionability of the profiles to support reproducibility, and was used to successfully re-execute the digital pathology annotation workflow described in Section 5.4.4.1 on page 171.

Of course, achieving a full re-execution in the general case may not always be possible: reproducibility is supported by the profiles, but also benefits from the characteristics of the workflow language (which should provide a clear formalism to map input items to their corresponding parameter slots) and from cooperation on the part of the workflow's author, who can help considerably by containerizing the environment required by each step and providing the relevant annotations (if allowed by the workflow language).

5.4.3.2 Galaxy

The Galaxy project [Galaxy 2022] provides a WfMS with data management functionalities as a multi-user platform, aiming to make computational biology more accessible to research scientists that do not have computer programming or systems administration experience. Galaxy's most prominent features include: a collection of 7500+ integrated tools[95]; a web interface that allows the execution and definition of workflows using the integrated tools; a network of dedicated (public) Galaxy instances.

The export of workflow execution provenance data as Workflow Run Crates has been added in Galaxy's 23.0 release. This feature provides a more interoperable alternative to the basic export of Galaxy workflow *invocations*: the workflow definition; a set of serialisations of the invocation-related metadata in Galaxy native, json-formatted files; and the input and output data files. This is achieved by extracting provenance from Galaxy entities related to the workflow run, along with associated metadata, converting them to RO-Crate metadata using the ro-crate-py library [De Geest 2023a]; by describing all files contained in the basic invocation export within the RO-crate metadata; and finally by making the Workflow Run Crate available for export to the user through Galaxy's web interface and API [De Geest 2022].

We extract the prospective provenance contained in Galaxy's YAML-based gxformat2[96] workflow definition, which includes details of the analysis pipeline such as the graph of tools that need to be executed, and metadata about the data types required. The retrospective provenance—i.e., the details of the executed workflow such as the inputs, outputs, parameter values used—is extracted from Galaxy's data model[97], which is not directly accessible to users in the context of a public Galaxy server. All of this provenance information is then mapped to RO-Crate metadata, including some Galaxy-specific data entities such as dataset collections. An exemplary exported Galaxy Workflow Run Crate is available on Zenodo [De Geest 2023b].

[95] https://galaxyproject.org/toolshed/

[96] https://galaxyproject.github.io/gxformat2/v19_09.html

[97] https://docs.galaxyproject.org/en/master/lib/galaxy.model.html

In practice, a user would take the following steps to obtain a Workflow Run Crate from a Galaxy instance:

(1) Create or download a Galaxy workflow definition (e.g.: from WorkflowHub) and import it in a Galaxy instance, or create a workflow through the Galaxy GUI directly.

(2) Execute the workflow, providing the required inputs.

(3) After the workflow has run successfully, the corresponding RO-Crate will be available for export from the Workflow Invocations list.

5.4.3.3 COMPSs

COMPSs [Lordan 2014] is a task-based programming model that allows users to transform a sequential application into a parallel one by simply annotating some of its methods, thus making it efficient to exploit the resources available (either distributed or in a cluster). When a COMPSs application is executed, a corresponding workflow describing the application's tasks and their data dependencies is dynamically generated and used by the COMPSs runtime to orchestrate the execution of the application in the infrastructure. As a WfMS, COMPSs stands out for its many advanced features that enable applications to achieve fine-grained high efficiency in HPC systems, such as the ability to exploit underlying parallelisation frameworks (i.e. MPI, OpenMP), compilers (e.g. NUMBA), failure management, task grouping, and more.

Provenance recording for COMPSs workflows has been explored in previous work [Sirvent 2022], where the Workflow RO-Crate profile was adopted in the implementation of a very lightweight approach to document provenance while avoiding the introduction of any significant run time performance overheads. However, because of the dynamic nature of COMPSs workflows, the Workflow Run Crate profile is better suited to represent them, since workflows are created when the application is executed and, thus, a prior static workflow definition does not exist before that moment. Due to this limitation, the workflow entity in the metadata file references the entry point application run by COMPSs, and formal parameters are not listed (note that listing them is not required by the profile) because inputs and outputs (both for each task and the whole workflow) are determined at runtime. COMPSs is able to export provenance data with a post-processing operation that can be triggered at any moment after the application has finished. The RO-Crate generation post-process uses information recorded by the runtime to detect and automatically add metadata of any input or output data assets used by the workflow.

Implementing Workflow Run Crate support in COMPSs required particular attention to the generation of a unique id for the *CreateAction* representing the workflow run, combining hostname and queuing system job id for supercomputer executions (as extra information added), and just using generated UUIDs for distributed environments, to add as much information as available from the run while ensuring the id is unique. In the *CreateAction*, the *description* term includes system information, as well as relevant environment variables that provide details on the execution environment (e.g., node list, CPUs per node). Finally, the *subjectOf* property of the *CreateAction* references the system's monitoring tool (when available), where authorised users can see detailed profiling of the corresponding job execution, as provided by the MareNostrum IV supercomputer[98].

To showcase the COMPSs adoption of the Workflow Run Crate profile, we provide as an example the execution of the BackTrackBB [Poiata 2016] application in the MareNostrum IV supercomputer. BackTrackBB targets the detection and location of seismic sources using the statistical coherence of the wave field recorded by seismic networks and antennas. The resulting RO-Crate [Poiata 2023] complies with the Workflow Run Crate profile, and includes the application source files, a diagram of the workflow's graph, application profiling and input and output files.

The implementation of provenance recording following Workflow Run Crate has been fully integrated in the COMPSs runtime, and is available since release 3.2[99] [Ejarque 2023].

5.4.3.4 StreamFlow

The StreamFlow[100] framework [Colonnelli 2021] is a container-native WfMS based on the CWL standard. It has been designed around two primary principles: first, it allows the execution of tasks in multi-container environments, supporting the concurrent execution of communicating tasks in a multi-agent ecosystem; second, it relaxes the requirement of a single shared data space, allowing for hybrid workflow executions on top of multi-cloud, hybrid cloud/HPC, and federated infrastructures. StreamFlow orchestrates hybrid workflows by combining a *workflow description* (e.g., a CWL workflow description and a set of input values) with one or more *deployment descriptions*—i.e. representations of the execution environments in terms of infrastructure-as-code (e.g., Docker Compose files [Reis 2022], HPC batch scripts, and Helm charts [Zerouali 2023]). A `streamflow.yml` file—the entry point of each StreamFlow execution—finds each workflow step with the set of most suitable execution environments. At execution time, StreamFlow automatically takes care of all the secondary aspects, like scheduling, checkpointing, fault tolerance, and data movements.

StreamFlow stores prospective and retrospective provenance data in a proprietary format into a persistent pluggable database (using sqlite3 as the default choice). After a CWL workflow execution completes, users can generate an RO-Crate through the `streamflow prov <workflow_name>` command, which extracts the provenance data stored in the database for one or more workflow executions and converts it to an RO-Crate archive that is fully compliant with the Provenance Run Crate Profile, including the details of each task run by the WfMS. Support for the format has been integrated into the main development branch and will be included in release 0.2.0 [Colonnelli 2023b].

From the StreamFlow point of view, the main limitation in the actual version of the Provenance Run Crate standard is the lack of support for distributed provenance, i.e., a standard way to describe the set of locations involved in a workflow execution and their topology. As a

[98] https://bsc.es/supportkc/docs/MareNostrum4/intro/
[99] https://github.com/bsc-wdc/compss/tree/3.2
[100] https://github.com/alpha-unito/streamflow

temporary solution, the StreamFlow configuration and a description of the hybrid execution environment are preserved by directly including the `streamflow.yml` file into the generated archive. However, this product-specific solution prevents a wider adoption from other WfMS and forces agnostic frameworks (e.g., WorkflowHub) to provide ad-hoc plugins to interpret the StreamFlow format. Since the support for hybrid and cross-facility workflows is gaining traction in the WfMS ecosystem, we envision support for distributed provenance as a feature for future versions of Workflow Run RO-Crate.

5.4.3.5 WfExS-backend

WfExS-backend[101] is a FAIR workflow execution orchestrator that aims to address some of the difficulties found in analysis reproducibility and analysis of sensitive data in a secure manner. WfExS-backend requires that the software used by workflow steps is available in publicly available software containers for reproducibility. Actual workflow execution is delegated to one of the supported workflow engines which matches with the workflow, right now either Nextflow or cwltool. The orchestrator prepares and stages all the elements needed to run the workflow—i.e. all the files of the workflow itself, the specific version of the workflow engine, the required software containers and the inputs. All these elements are referred through resolvable identifiers, ideally public, permanent ones. Due to this, the orchestrator can consume workflows which are originally available in different kinds of locations, like git repositories, Software Heritage, or even RO-Crates from WorkflowHub.

WfExS-backend development milestones aim to reach FAIR workflow execution through the generation and consumption of RO-Crates following the latest Workflow Run Crate profiles, which have proven to be a mechanism suitable to semantically describe digital objects in a way that simplifies embedding key details involved in analysis reproducibility and replicability.

The orchestrator records details relevant to the prospective provenance when a workflow is prepared for execution, such as the public URLs used to fetch input data and workflows, content digestion fingerprints (typically sha256 checksums) and metadata derived from workflow files, container images and input files. Most of this metadata is represented in the generated RO-Crates. WfExS-backend has explicit commands to generate and publish both prospective and retrospective provenance RO-Crates based on a given existing staged execution scenario. These RO-Crates can selectively include copies of used elements as payloads. Workflows can be executed more than once in the same staged directory, with all the executions sharing the same inputs. Thus, run details from all the executions are represented in the retrospective provenance RO-Crate. Support for Workflow Run RO-Crate is available since WfExS-backend version 0.10.1 [Fernández 2023a]. Future developments will also add support for embedding URLs of output results that have been deposited into a suitable repository (like Zenodo DOIs, for instance) as well as consuming previously produced RO-Crates.

An example of Workflow Run Crate generated by WfExS-backend from a Nextflow workflow

[101] https://github.com/inab/WfExS-backend

run [Bouyssié 2023] is available from Zenodo [Fernández 2023b].

5.4.3.6 Sapporo

Sapporo [Suetake 2022] is an implementation of the Workflow Execution Service (WES) API specification[102]. WES is a standard proposed by the Global Alliance for Genomics and Health (GA4GH) for cloud-based data analysis platforms that receive requests to execute workflows. Sapporo supports the execution of several workflow engines, including cwltool [Amstutz 2023], Toil [Vivian 2017], and StreamFlow [Colonnelli 2021]. Sapporo includes features specifically tailored to bioinformatics applications, including the calculation of feature statistics from specific types of outputs generated by workflow runs. For example, the system calculates the mapping rate of DNA sequence alignments from BAM format files. To describe the feature values, Sapporo uses the Workflow Run Crate profile extended with additional terms to represent these biological features[103].

Further, the Tonkaz companion command line software has integrated functionality to compare Run Crates generated by Sapporo to measure the reproducibility of the workflow outputs [Suetake 2023a]. Developers can use this unique feature to build a CI/CD platform for their workflows to ensure that changes to the product do not produce an unexpected result. Workflow users can also use this feature to verify the results from the same workflow deployed in different environments.

While Sapporo supports Workflow Run Crate, since WES is a WfMS wrapper, it does not parse the provided workflow definition files. Instead, it embeds the information in the files passed by the WES request to record the provenance of execution rather than using the actual workflow parameters meant for the wrapped WfMS. Therefore, the current implementation of Sapporo does not capture the connections between the inputs/outputs depicted in the workflow and the actual files used/generated during the run. Thus, the profile generated by Sapporo has fields representing input and output files, but they are not linked to formal parameters.

Sapporo supports export to Workflow Run Crate since release 1.5.1 [Suetake 2023b]. An example of RO-Crate generated by Sapporo is available on Zenodo [Ohta 2023].

5.4.3.7 Autosubmit

Autosubmit [Manubens-Gil 2016] is an open source lightweight workflow manager and meta-scheduler created in 2011 for use in climate research to configure and run scientific experiments. It supports scheduling jobs via SSH to Slurm [Yoo 2003], PBS [Feng 2007] and other remote batch servers used in HPC.

The “archive” feature was added in Autosubmit 3.1.0[104], released in 2015. This feature archives the experiment directory and all its contents into a ZIP file, which can be used later to access the

[102] https://www.ga4gh.org/product/workflow-execution-service-wes/
[103] https://github.com/ResearchObject/ro-terms/tree/master/sapporo
[104] https://earth.bsc.es/gitlab/es/autosubmit/-/tags/v3.1.0

provenance data or to execute the Autosubmit experiment again. Even though the data in the ZIP file includes prospective provenance and retrospective provenance, it contains no structure, and users have no way to tell which is which from just looking at the ZIP file and its contents.

Recent releases of Autosubmit 4 include an updated YAML configuration management system that allows users to combine multiple YAML files into a single unified configuration file. While this gave users flexibility, it also increased the complexity to track the configuration changes and to relate these to the provenance data. Another feature added in Autosubmit 4 is the option to use only the experiment manager features of Autosubmit, delegating the workflow execution to a different backend workflow engine, like ecFlow [Bahra 2011], Cylc [Oliver 2019], or a CWL runner.

In order to give users a more structured way to archive provenance, which includes the complete experiment configuration and the parameters used to generate the unified experiment configuration, and also to allow interoperability between workflow managers, the archive feature received a new flag in Autosubmit 4.0.100 [Beltrán 2023] to generate Workflow Run RO-Crates.

The prospective provenance data is extracted from the Autosubmit experiment configuration. This includes the multiple YAML files, and the unified YAML configuration, as well as the parameters used to preprocess each file—preprocessing replaces placeholders in script templates with values from the experiment configuration. The retrospective provenance data is included with the RO-Crate archive and includes logs and other traces produced by the experiment workflow. Both prospective and retrospective provenance data are included in the final RO-Crate JSON-LD metadata file. Autosubmit uses the Workflow Run RO-Crate profile.

As one of the most recent implementations, much of the code added in Autosubmit 4 for RO-Crates was adapted from existing implementations like COMPSs and StreamFlow. ro-crate-py [De Geest 2023a] was used for the heavy lifting work of creating the RO-Crate archive in Python, and adding information for the JSON-LD metadata.

The main challenges for adopting RO-Crate in Autosubmit were incorporating Autosubmit's "Project" feature, and the lack of validation tools and of documentation and examples on how to use the standard with *coarse-grained* workflow management systems (as described in [Goble 2020]) that do not track inputs and outputs, which is the case of Autosubmit—as well as the Cylc and ecFlow workflow engines.

A Project in Autosubmit is an abstract concept that has a type and a location, and is used to separate experiment configuration and template scripts and other auxiliary files The type can be Git, Subversion, or Local. For each type the location represents the URL of a code repository, or a directory on a workstation or HPC file system used to copy the Project and its template scripts (written in Shell, R, or Python) and any other files (input data for a model, extra configuration files, binaries, etc.). The workflows in Autosubmit have tasks with dependencies to other tasks, and each of these tasks execute one of these template scripts. The RO-Crate file generated by Autosubmit includes only the project type and location, and not the complete Project. Therefore, users have the provenance of the Project, but only those with the correct permissions can access its constituent resources (many applications run with Autosubmit can not be publicly shared without consent).

Validation tools for RO-Crate archives are still under development, and while there is a community-based review process to help and guide new implementations, a tool that others can use as code is written will contribute to a more agile development.

After working with the RO-Crate community on these issues, the Autosubmit team adopted a mixed approach where Autosubmit initialises the JSON-LD metadata from its configuration and local trace files, and the user is responsible for providing a partial JSON-LD metadata object in the Autosubmit YAML configuration. A pull request was created to ro-crate-py to allow the RO-Crate JSON-LD metadata to be patched by these partial JSON-LD metadata objects. This way, users are able to provide the missing information from the Autosubmit configuration model, like licence, authors, inputs, outputs, formal parameters, and more. And by modifying ro-crate-py, future implementers of RO-Crate that have a similar workflow configuration as Autosubmit should be able to re-use it, while also using COMPSs, StreamFlow, Autosubmit, and other implementations as reference.

A workflow was created using an example Autosubmit Project [Kinoshita 2023] designed using UFZ's mHM (mesoscale Hydrological Model) [Samaniego 2010, Kumar 2013]. This workflow was used to validate the RO-Crate produced by Autosubmit. This validation was performed by the Workflow Run RO-Crate community in a public GitHub repository[105] and also using the aforementioned Runcrate.

5.4.3.8 Summary of implementations

Table 5.1 on the following page shows an overview of the different implementations presented in this section.

5.4.4 Exemplary Use Cases

We illustrate Workflow Run RO-Crate on two exemplary use cases, which are similar in terms of application domain—machine learning-aided tumour detection in human prostate images—but quite different in the way computations are executed and provenance is represented: in the first, the analysis is conducted by means of a CWL workflow and the outcome is represented with Provenance Run Crate; in the second, a combination of Process Run Crate and CPM RO-Crate is used to represent a sequence of computations linked to their corresponding CPM provenance information.

5.4.4.1 Provenance Run Crate for Digital Pathology

We present a use case that demonstrates the effectiveness of our most detailed profile Provenance Run Crate at recording provenance data in the context of digital pathology. More specifically, we

[105] https://github.com/ResearchObject/workflow-run-crate/

Impl.	Profile	Version URL/DOI	Example
runcrate	Provenance	[Leo 2023a]	[Leo 2023c]
Galaxy	Workflow	[Afgan 2023]	[De Geest 2023b]
COMPSs	Workflow	[Ejarque 2023]	[Poiata 2023]
Streamflow	Provenance	[Colonnelli 2023b]	[Colonnelli 2023a]
WfExS	Workflow	[Fernández 2023a]	[Fernández 2023b]
Sapporo	Workflow	[Suetake 2023b]	[Ohta 2023]
Autosubmit	Workflow	[Beltrán 2023]	[Kinoshita 2023]

Table 5.1: Workflow Run Crate implementations. Summary of each WRROC implementation, together with the profiles it implements, the latest software citation and an example crate of its application. Runcrate is a toolkit that converts CWLProv ROs to Provenance Run Crates, while the others are WfMS.

demonstrate the generation of RO-Crates to save provenance data associated with the computational annotation of magnified prostate tissue areas and cancer subregions using deep learning models [Del Rio 2022]. The image annotation process is implemented in a CWL workflow consisting of three steps, each executing inference on an image using a deep learning model: inference of a low-resolution tissue mask to select areas for further processing; high-resolution tissue inference on areas identified in the previous step to refine borders; high-resolution cancer identification on areas identified in the first step. The two tissue inference steps run the same tool, but set different values for the parameter that controls the magnification level. The workflow is integrated in the CRS4 Digital Pathology Platform[106], a web-based platform to support clinical studies involving the examination and/or the annotation of digital pathology images.

To assess the interoperability of WRROC, we recorded the provenance of the same exemplary workflow in two different execution platforms. In the first case, the workflow was executed with the StreamFlow WfMS, for which the Provenance Run Crate implementation is discussed in Section 5.4.3.4 on page 167. In the second case, we executed the CWL workflow with cwltool and converted the resulting CWLProv RO to a Provenance Run Crate with the runcrate tool (Section 5.4.3.1 on page 164).

The RO-Crates obtained in the two cases [Colonnelli 2023a, Leo 2023c] are very similar to each other, differing only in a few details: for instance, [Colonnelli 2023a] includes the StreamFlow configuration file and has separate files for the workflow and the two tools, while [Leo 2023c] has the workflow and the tools stored in a single file (CWL's "packed" format). Apart from these minor differences, the description of the computation is essentially the same.

Four actions are represented: the workflow itself, the two executions of the tissue extraction tool and the execution of the tumour classification tool. Each action is linked to the corresponding

[106] https://github.com/crs4/DigitalPathologyPlatform

workflow or tool via the *instrument* property, and reports its starting and ending time. For each action, input and output slots are referenced by the workflow, while the corresponding values are referenced by the action itself. The data entities and *PropertyValue* instances corresponding to the input and output values link to the corresponding parameter slots via the *exampleOfWork* property, providing information on the values taken by the parameters.

Listing 5.4 shows the output of the `runcrate report` command for the StreamFlow RO-Crate. For each action (workflow or tool run), the tool reports the associated instrument (workflow or tool), the starting and ending time and the list of inputs and outputs, with arrows pointing from the formal parameter to the corresponding actual value taken during the execution of the action.

Listing 5.4: `runcrate report` command line output. This informal listing of relevant RO-Crate entities describe each step execution. Note that inputs and outputs are of different types (not shown), e.g. `tissue_low>0.9` is a string parameter, `6b15de`… is a filename, and `#af0253`… is a collection.

```
action: #30a65cba-1b75-47dc-ad47-1d33819cf156
  instrument: predictions.cwl (['SoftwareSourceCode',
         'ComputationalWorkflow', 'HowTo', 'File'])
  started: 2023-05-09T05:10:53.937305+00:00
  ended: 2023-05-09T05:11:07.521396+00:00
  inputs:
    #af0253d688f3409a2c6d24bf6b35df7c4e271292 <- predictions.cwl#slide
    tissue_low <- predictions.cwl#tissue-low-label
    9 <- predictions.cwl#tissue-low-level
    tissue_low>0.9 <- predictions.cwl#tissue-high-filter
    tissue_high <- predictions.cwl#tissue-high-label
    4 <- predictions.cwl#tissue-high-level
    tissue_low>0.99 <- predictions.cwl#tumor-filter
    tumor <- predictions.cwl#tumor-label
    1 <- predictions.cwl#tumor-level
  outputs:
    06133ec5f8973ec3cc5281e5df56421c3228c221 <- predictions.cwl#tissue
    4fd6110ee3c544182027f82ffe84b5ae7db5fb81 <- predictions.cwl#tumor
action: #457c80d0-75e8-46d6-bada-b3fe82ea0ef1
  step: predictions.cwl#extract-tissue-low
  instrument: extract_tissue.cwl (['SoftwareApplication', 'File'])
  started: 2023-05-09T05:10:55.236742+00:00
  ended: 2023-05-09T05:10:55.910025+00:00
  inputs:
    tissue_low <- extract_tissue.cwl#label
    9 <- extract_tissue.cwl#level
    #af0253d688f3409a2c6d24bf6b35df7c4e271292 <- extract_tissue.cwl#src
  outputs:
    6b15de40dd0ee3234062d0f261c77575a60de0f2 <- extract_tissue.cwl#tissue
action: #d09a8355-1a14-4ea4-b00b-122e010e5cc9
  step: predictions.cwl#extract-tissue-high
  instrument: extract_tissue.cwl (['SoftwareApplication', 'File'])
  started: 2023-05-09T05:10:58.417760+00:00
  ended: 2023-05-09T05:11:03.153912+00:00
  inputs:
    tissue_low>0.9 <- extract_tissue.cwl#filter
    6b15de40dd0ee3234062d0f261c77575a60de0f2 <- extract_tissue.cwl#filter_slide
    tissue_high <- extract_tissue.cwl#label
    4 <- extract_tissue.cwl#level
    #af0253d688f3409a2c6d24bf6b35df7c4e271292 <- extract_tissue.cwl#src
  outputs:
    06133ec5f8973ec3cc5281e5df56421c3228c221 <- extract_tissue.cwl#tissue
action: #ae2163a8-1a2a-4d78-9c81-caad76a72e47
  step: predictions.cwl#classify-tumor
  instrument: classify_tumor.cwl (['SoftwareApplication', 'File'])
  started: 2023-05-09T05:10:58.420654+00:00
  ended: 2023-05-09T05:11:06.708344+00:00
  inputs:
    tissue_low>0.99 <- classify_tumor.cwl#filter
    6b15de40dd0ee3234062d0f261c77575a60de0f2 <- classify_tumor.cwl#filter_slide
    tumor <- classify_tumor.cwl#label
    1 <- classify_tumor.cwl#level
    #af0253d688f3409a2c6d24bf6b35df7c4e271292 <- classify_tumor.cwl#src
  outputs:
    4fd6110ee3c544182027f82ffe84b5ae7db5fb81 <- classify_tumor.cwl#tumor
```

The *exampleOfWork* link between input / output values and parameter slots is used by `runcrate run` to reconstruct the CWL input parameters document needed to rerun the computation. The *alternateName* property (a Schema.org property applicable to all entities), which records the original name of data entities (at the time the computation was run), is also crucial for reproducibility in this case: both StreamFlow and CWLProv, to avoid clashes, record input and output files and directories using their SHA1 checksum as their names. However, this particular workflow expects the input dataset to be in the MIRAX[107] format, where the "main" file taken as an input parameter by the processing application must be accompanied by a directory in the same location with the same name apart from the extension. The runcrate tool uses the *alternateName* to rename the input dataset as required, so that the expected pattern can be picked up by the workflow during the re-execution. This use case was the main motivation to include a recommendation to use *alternateName* with the above semantics in Process Run Crate.

Thanks to the fact that both RO-Crates were generated following the best practices to support reproducibility mentioned in the profiles, we were able to re-execute both computations with the runcrate tool. This was also made possible by the fact that the CWL workflow included information on which container images to use for each tool. Overall, this shows how reproducibility is a hard-to-achieve goal that can only be supported, but not ensured, by the profiles, since it also depends on factors like the characteristics of the computation, the choice of workflow language and whether best practices such as containerisation are followed.

This use case highlighted the need to add specifications on how to represent multi-file datasets [WRROC 2023a, section Representing multi-file objects]. In the MIRAX format, in fact, the "main" file must be accompanied by a directory in the same location containing additional files with a specific structure. To represent this, we added specifications to the Process Run Crate

[107] https://openslide.org/formats/mirax/

profile on describing "composite" datasets[108] consisting of multiple files and directories to be treated as a single unit—as opposed to more conventional input or output parameters consisting of a single file. The profile specifies that such datasets should be represented by a *Collection* entity linking to individual files and directories via the *hasPart* property, and referencing the main part (if any) via the *mainEntity* property. Note that, by adding this specification to Process Run Crate, we also made it available to Workflow Run Crate and Provenance Run Crate. In the output of the runcrate report tool the additional files are not shown, since the formal parameter points to the *Collection* entity that describes the whole dataset.

5.4.4.2 Process Run Crate and CPM RO-Crate for cancer detection

This section presents an RO-Crate created to describe an execution of a computational pipeline that trains AI models to detect the presence of carcinoma cells in high-resolution digital images of magnified human prostate tissue. The RO-Crate makes use of Process Run Crate and CPM RO-Crate[109], an RO-Crate profile that supports the representation of entities described according to the Common Provenance Model (CPM) [Wittner 2022, Wittner 2023b].

The CPM, an extension of the W3C PROV model [Moreau 2013] is a recently developed provenance model that enables the representation of distributed provenance. Distributed provenance is created when an object involved in the research process, either digital or physical (e.g., biological material), is exchanged between organisations, so that each organisation can document only a portion of the object's life cycle. Individual provenance components are generated, stored, and managed individually by each organisation, and are linked together in a chain. The CPM prescribes how to represent such provenance, and how to enable its traversal and processing using a common algorithm, independently from the type of object being described. In addition, the CPM defines a notion of meta-provenance, which contains metadata about the history of individual provenance components.

CPM RO-Crate supports the identification of CPM-based provenance and meta-provenance files within an RO-Crate, allowing to pack data, metadata, and CPM-based provenance information together. An RO-Crate generated according to the CPM-RO-Crate profile embeds parts of the distributed provenance, which may be linked to the provenance of precursors and successors of the packed data.

The CPM-RO-Crate profile synergises well with Process Run Crate, since the former can add references to CPM-based provenance descriptions of computational executions described with the latter, integrating them in the distributed provenance. Since CPM-based provenance and meta-provenance files are typically themselves produced by computations, Process Run Crate allows to represent these along with the main computations that produce the datasets being exchanged, providing the full picture in a cohesive ensemble.

The pipeline consists of three main computational steps: a preprocessing step that splits input

[108] https://w3id.org/ro/wfrun/process/0.4#representing-multi-file-objects
[109] https://w3id.org/cpm/ro-crate

images into small patches and divides them into a training and a testing set; a training step that trains the model to recognise the presence of carcinoma cells in the images; an evaluation step that measures the accuracy of the trained model on the testing set. In addition to the pipeline steps, the RO-Crate describes additional computations related to the generation of the CPM provenance and meta-provenance files. All computations are described according to the Process Run Crate profile, while the CPM files are referenced according to the CPM RO-Crate profile.

Also represented via Process Run Crate are: the input dataset; the results of the pipeline execution; the scripts that implement the pipeline; the log files generated by the scripts; a script that converts the logs into the CPM files. This allowed us to describe all involved elements as a single aggregate, with entities and their relationships represented according to the RO-Crate model. The RO-Crate discussed here is available from Zenodo [Wittner 2023c].

The CPM files complement the RO-Crate with internal details about the pipeline execution, such as how the input dataset was split into training and testing sets, or detailed information about each training iteration of the AI model. For instance, it contains a representation of a checkpoint of the AI model after the second training iteration. The corresponding entity's attributes contain paths to the respective model stored as a file. The entity is related to the respective training iteration activity, which contains the iteration parameters represented as an attribute list.

In addition, the CPM generally provides means to link the input dataset provenance to the provenance of its precursors—human prostate tissues and biological samples the tissues were derived from; this is not included in the example because we used a publicly available input database for which provenance of the precursors was not available. However, the linking mechanism for provenance precursors is exactly the same as between the bundles for the AI pipeline parts.

While the RO-Crate is focused on the execution of the pipeline, the provenance included in the CPM files intends to be interlinked with provenance of the precursors or successors, providing means to traverse the whole provenance chain. For the described digital pathology pipeline, the precursors would be:

(1) A biological sample acquired from a patient.

(2) Slices of the sample processed and put on glass slides.

(3) The images created as a result of scanning the slides using a microscope.

As a result, combining the CPM and RO-Crate enables the lookup of research artefacts related to the computation across heterogeneous organisations using the underlying provenance chain.

5.4.5 Discussion

The RO-Crate profiles presented here provide a unified data model to describe the prospective and retrospective provenance of the execution of a computational workflow, together with contextual metadata about the workflow itself and its associated entities (inputs, outputs, code, etc.). The profiles are flexible, allowing to tailor the description to a broad variety of use cases, agnostic with respect to the WfMS used and allow describing provenance traces at different levels of granularity. This facilitates developing implementations by multiple workflow systems (often with heterogeneous assumptions and requirements)—six of which have already been developed and are described in Section 5.4.3 on page 162—allowing to perform comparisons between runs across heterogeneous systems. For instance, the SPARQL[110] query in Listing 5.5 returns all actions in a Workflow Run RO-Crate, together with their instruments and their starting and ending times:

```
PREFIX schema: <http://schema.org/>
SELECT ?action ?instrument ?start ?end
WHERE {
  ?action a schema:CreateAction .
  ?action schema:instrument ?instrument .
  OPTIONAL { ?action schema:startTime ?start } .
  OPTIONAL { ?action schema:endTime ?end }
}
```

Listing 5.5: SPARQL query to find actions in a Workflow Run RO-Crate

Additionally, having workflow runs and plans described according to the RO-Crate model allows capturing the context of the workflow itself (e.g. authors, related publications, other workflows, etc.) rather than the trace alone. Being based on RO-Crate, the profiles and their implementations are part of a growing ecosystem[111] of tools and services maintained by the RO-Crate community.

Another advantage of RO-Crate is that the files corresponding to the data entities (inputs, outputs, code, etc.) do not necessarily have to be stored together with the metadata file: for instance, they can be remote and referred to via an http(s) URI. This is mostly relevant in situations where the file is very large or cannot be shared publicly: the data entity's identifier can be a URI that is accessible only through authentication, or resolvable only within the boundaries of the generating organisation.

The derivation of Workflow Run Crate from Workflow RO-Crate and, in turn, of Provenance Run Crate from Workflow Run Crate makes RO-Crates that conform to these profiles compatible with the WorkflowHub workflow registry, allowing workflow runs to be registered and easily found and shared with other researchers. Additionally, the inheritance mechanism allows reusing the specifications already developed for Workflow RO-Crate, which form part of the guidelines on representing the prospective provenance

The Workflows Community Summit [Ferreira da Silva 2023] identified as one of the current open challenges in the Scientific Workflows domain the ability to build FAIR into Workflow Management Systems, with the objective of achieving FAIR Computational Workflows. The pro-

[110] https://www.w3.org/TR/sparql11-overview/

[111] https://www.researchobject.org/ro-crate/in-use/

files introduced in this article are able to help tackle this by introducing interoperable metadata among WfMSs that captures the provenance of their corresponding workflow executions.

The Workflow Run RO-Crate profiles, the associated tooling, the implementations and the examples are developed by a community that runs regular virtual meetings (every two weeks at the time of writing) and coordinates on Slack and the RO-Crate mailing list. The WRROC community brings together members of the RO-Crate community [Soiland-Reyes 2022a], WfMS users and developers, Workflow users and developers, GA4GH [Rehm 2021] Cloud developers and provenance model authors, and is open to anyone who is interested in the representation of workflow provenance. The inclusion of WfMS developers and workflow users was key to keeping the specifications flexible, easy to implement and grounded on real use cases, while the diversity of the stakeholders allowed to keep a plurality of viewpoints while driving the model's development forward.

One of the main benefits of this development process is that the profiles are already in use, with seven implementations (six WfMS and one conversion tool) already available as described in Section 5.4.3 on page 162.

In the following subsections, we provide an evaluation of the metadata coverage of runcrate and we discuss WRROC relates to standards such as W3C PROV and to other community projects.

5.4.5.1 Evaluation of metadata coverage using runcrate convert

In order to assess the metadata coverage of *Leo 2023a* (Section 5.4.3.1 on page 164), we performed a qualitative analysis of the tool's *convert* mode, in which we evaluated how the generated RO-Crates preserve the metadata contained in the CWLProv ROs from which they are derived. For this analysis, we followed the same approach as for an earlier evaluation of CWLProv [de Wit 2022]. In that work, we identified and analysed three levels of representation: firstly, in RDF; secondly, in a structured, but CWL-specific document; and finally, in an unstructured, human-readable format. From this earlier analysis, we concluded that the CWLProv RDF representation of the workflow runs lacked many provenance metadata that was included in CWL-specific documents, such as the packed workflow and input parameter file. For example, the CWLProv RDF only contained the name of each workflow step, without including the link to the underlying CommandLineTool or nested Workflow that was executed; information that could be extracted from the packed workflow.

In our analysis of runcrate, we compared the CWLProv RDF provenance graph with the RO-Crate metadata file. The results of the analysis are summarised in Table 5.2 on page 180 [112]. Overall, most of the information contained in CWLProv RDF is transferred to the RO-Crate metadata. In addition, the representation of some categories of metadata has improved, notably Workflow parameters (WF2), which were insufficiently described in CWLProv RDF but defined

[112] The three dots (...) in the WRROC column indicate that the concept is supported in an RO-Crate using existing schema.org vocabulary (e.g. https://schema.org/softwareHelp) but is not required or recommended by the WRROC profiles.

with type and format in RO-Crate. Moreover, the format of input files (D2), which was partially represented in CWLProv RDF, is fully represented in RO-Crate.

In conclusion, our analysis shows that runcrate preserves most provenance metadata previously shown to be relevant in realistic RO use case scenarios. The full results of the analysis can be found in [de Wit 2023].

From this analysis it is worth highlighting the gaps and potential for Workflow Run RO-Crates. Several areas have been flagged by this study as important aspects of workflow metadata, such as Data Access (D3), Software Documentation (SW2) and Workflow Requirements (WF3). Many such aspects require human annotation and cannot be provided by workflow engines alone, although they may be propagated from workflow and tool definitions. Some areas like Consumed Resources (EX2) require additional terms to be defined, and are part of future work.

5.4.5.2 Workflow Run RO-Crate and the W3C PROV standard

Our aim is to be compatible with both Schema.org and W3C PROV. Provenance Run Crate is the profile that most closely matches the level of detail provided by CWLProv, which extends W3C PROV. Table 5.3 on page 181 shows how the main entities and relationships represented by Provenance Run Crate map to PROV constructs, using the SKOS vocabulary to indicate the type of relationship between each pair of terms. A machine-readable version of the mapping can be found in the accompanying RO-Crate[113] of this article [Leo 2023b].

5.4.5.3 Five Safes Workflow Run Crate

The *Five Safes RO-Crate* [Soiland-Reyes 2023d] profile has been developed to extend the Workflow Run RO- Crate profile for use in Trusted Research Environments (TRE) in order to handle sensitive health data in federated workflow execution across TREs in the UK [Giles 2023] and following the Five Safes Framework [Desai 2016]. A crate with a workflow run request references a pre-approved workflow and project details for manual and automated assessment according to the TRE's agreement policy for the sensitive dataset.

The crate then goes through multiple phases internal to the TRE, including validation, sign-off, workflow execution and disclosure control [Soiland-Reyes 2023e]. At this stage the crate is also conforming to the Workflow Run Crate profile. The final crate is then safe to be made public. This extension of Workflow Run Crate documents and supports the *human review process*—important for transparency on TRE data usage. The initial implementation of this profile used WfExS as the workflow execution backend, and this approach will form the basis for further work on implementing federated workflow execution in the British initiatives DARE UK and HDR UK[114] [Snowley 2023] and in the European EOSC-ENTRUST[115] project for Trusted Research Environment.

[113] https://w3id.org/ro/doi/10.5281/zenodo.10368989
[114] https://esciencelab.org.uk/projects/federated-analytics/
[115] https://esciencelab.org.uk/projects/eosc-entrust/

Table 5.2: Summarised results of our qualitative analysis of runcrate

Type	Subtype	Name	CWL	CWLProv	RO-Crate	WRROC
T1	SC1	Workflow design	●	·	○	…
	SC2	Entity annotations	·	·	·	…
	SC3	Workflow execution ann.	·	·	·	…
T2	D1	Data identification	○	·	·	…
	D2	File characteristics	○	○	●	○
	D3	Data access	○	·	·	…
	D4	Parameter mapping	●	●	●	●
T3	SW1	Software identification	○	·	○	…
	SW2	Software documentation	·	·	·	…
	SW3	Software access	·	·	·	…
T4	WF1	Workflow software	●	○	○	…
	WF2	Workflow parameters	●	○	●	●
	WF3	Workflow requirements	●	·	○	○
T5	ENV1	Software environment	·	·	·	·
	ENV2	Hardware environment	·	·	·	·
	ENV3	Container image	○	○	○	●
T6	EX1	Execution timestamps	·	●	●	●
	EX2	Consumed resources	·	·	·	·
	EX3	Workflow engine	·	○	○	○
	EX4	Human agent	·	●	●	●

We compared RO-Crates with the CWLProv ROs from which they were generated. The analysis was based on a provenance taxonomy reflecting relevant provenance metadata based on realistic use cases for ROs associated with a real-life bioinformatics workflow [de Wit 2022]. CWL-specific documents are: `packed.cwl` (the workflow), `primary-job.json` (the inputs file), and `primary-output.json` (the outputs file). Since `packed.cwl` is also included in RO-Crate, we only considered how the metadata was represented in `ro-crate-metadata.json`.

For completeness we also show the theoretical capability of the Provenance Run Crate profile (WRROC column) assuming all its MUST/SHOULD requirements are complete. The categories in the first three columns are explained in [de Wit 2022].

Legend: ● fully represented ○ partially represented · missing or unstructured representation … optional (e.g. schema.org attribute)

Table 5.3: Mapping from Workflow Run RO-Crate to equivalent W3C PROV concepts using SKOS [Isaac 2009]. For instance, *CreateAction* has **broader** match PROV's *Activity*, meaning that *Activity* is more general.

RO-Crate	**Relationship**	**W3C PROV-O**
Action (superclass of *CreateAction*, *OrganizeAction*)	Has close match (schema.org Actions may also be potential actions in the future)	*Activity*
CreateAction, *OrganizeAction*	Has broader match	*Activity*
Person	Has exact match	*Person*
Organization	Has exact match	*OrganizeAction*
SoftwareApplication	Has related match	*SoftwareAgent*
ComputationalWorkflow, *SoftwareApplication*, *HowTo*	Has broader match	*Plan*, *Entity*
File, *Dataset*, *PropertyValue*	Has broader match	*Entity*
startTime on *CreateAction*	Has close match	*startedAtTime*
endTime on *CreateAction*	Has close match	*endedAtTime*
agent on *CreateAction*	Has related match	*wasStartedBy*, *wasEndedBy*
agent and *instrument* on *CreateAction*	Has broader match	*wasAssociatedWith*
instrument on *CreateAction*	Has related match (Complex mapping: an instrument implies a qualified association with the agent, linked to a plan)	*hadPlan* on *Association*
object on *CreateAction*	Has exact match	*used*
result on CreateAction	Has close match	inverse *wasGeneratedBy*

5.4.5.4 Biocompute Object RO-Crate

[IEEE 2791-2020], colloquially *Biocompute Objects* (BCO), is a standard for representing provenance of a genomic sequencing pipeline, intended for submission of the workflow to regulatory bodies, e.g. as part of a personalised medical treatment method [Alterovitz 2018]. The BCO is represented as a single JSON file which includes description of the workflow and its steps and intended purpose, as well as references for tools used and data sources accessed. There is overlap in the goals of BCO and Workflow Run Crate profiles; however, their intentions and focus are different. BCO is primarily conveying a computational method for the purpose of manual regulatory review and further reuse, with any values provided as an exemplar run. A Workflow Run Crate, however, is primarily documenting a particular workflow execution, and the workflow is associated to facilitate rerun rather than reuse.

Previously, a guide[116] to packaging BioCompute Objects using RO-Crate was developed as a profile to combine both standards [Soiland-Reyes 2021]. In this early approach, RO-Crate was primarily a vessel to transport the BCO along with its constituent resources, including the workflow and data files, as well as provide these resources with additional typing and licence metadata that is not captured by the BCO JSON. Further work is being planned with the BCO community to update the BCO-RO profile to align with the newer Workflow Run Crate profiles.

5.4.6 Conclusion and Future Work

In this work we presented Workflow Run RO-Crate, a collection of RO-Crate profiles to represent the provenance of the execution of computational workflows at different levels of granularity. We described each profile and their corresponding implementations, shown how they apply to real use cases and described the community behind their development process. Workflow Run RO-Crate has already been adopted by six WfMS, including Galaxy, StreamFlow and COMPSs. The flexibility of our model eases its implementation in more systems, allowing interoperability between their workflow run descriptions.

Workflow Run RO-Crate is an ongoing project driven by an open community. A natural consequence of this is that the profiles are not static entities, but keep being updated to cater for new requirements and use cases. In-progress features are tracked in the GitHub repository issues[117] and are open to discussion for the community. New features under discussion include a representation of the execution environment and recording workflow resource usage. The runcrate toolkit is planned to be expanded both to better support the current features and to include new ones that may arise.

Many of the presented implementations will also develop new features. For example, the Galaxy implementation will add metadata detailing each step of a workflow run to conform to the Provenance Run Crate profile; develop and/or integrate RO-Crate more deeply with import and export of Galaxy histories through the implementation of a profile; and further

[116] https://biocompute-objects.github.io/bco-ro-crate/

[117] https://github.com/ResearchObject/workflow-run-crate/issues

developing features to allow for user-guided import of RO-Crates as Galaxy datasets, histories and workflows.

Finally, we are currently exploring the cloud execution of Workflow Run RO-Crates. On the one hand, the Workflow Execution Service (WES) specification is used by the Global Alliance for Genomics and Health (GA4GH) [Rehm 2021] to enable WfMS-agnostic interpretation of workflows and scheduling of task execution. On the other hand, the Task Execution Service (TES) specification enables the execution of individual, atomic, containerised tasks in a compute backend-independent manner.

We are planning to undertake an in-depth analysis of the degree of interoperability between the TES and WES API standards—roughly the equivalents of Process and Workflow Run Crates, respectively—by placing their focus on the actual execution of tasks/processes and workflows in cloud environments and liaising with the GA4GH Cloud community to align schemas where necessary. We will then build an interconversion library that attempts to:

(1) Construct WES workflow and TES task run requests from RO-Crates containing Provenance, Workflow or Process Run requests and therefore allow their easy (re)execution on any GA4GH Cloud API-powered infrastructure.

(2) Bundle information from the WES and TES (as well as other GA4GH Cloud API resources, where available) to create or extend RO-Crates with standards-compliant Process, Workflow or even Provenance RO-Crates.

6

Discussion and conclusions

6.1 Discussion

In this section I summarise and discuss the findings from the previous chapters, relating them to emerging related work and future directions.

6.1.1 Making a predictable ecosystem of FAIR digital objects

The main advantage of scholarly researchers publishing FAIR data is to enable machine actionability [Wilkinson 2016], which again will accelerate further research, such as through computational workflows. In practice, data publishing is largely approached either by depositions in general and institutional repositories for Open Data such as Figshare and Zenodo [Dillen 2019a], or to specialised domain-specific repositories such as in biodiversity [GBIF 2021].

European research infrastructures supporting Open Science practices are coalescing their services to form the European Open Science Cloud (EOSC) [Ayris 2016], which are embracing FAIR principles [Mons 2017] and building a common framework for interoperability [Kurowski 2021].

While existing practices for implementing FAIR have relied on the Linked Data (LD) stack, that is just one possible technology to achieve the benefits of interoperable machine actionability [Mons 2017].

Chapter 3 explored the emerging concept of *FAIR Digital Objects* (FDO) [Schultes 2019] as a potential distributed object system for FAIR data, comparing its proposed principles and current practices with the established Linked Data approach. As detailed in Section 2.1 on page 16, FDO defines a handful of constraints and guides for a predictable way to organise complex machine actionable digital entities.

Conceptually FDO can clearly be useful for realizing FAIR principles with more active digital objects that can form a consistent ecosystem, but this opens many questions on actual FDO implementations with regards to protocols and standards.

6.1.1.1 Linked Data need more constraints and consistency to be FAIR

Examined in Section 2.2 on page 23, the principles of *Linked Data* emerged from the Semantic Web as a data-centric view with a focus on navigation and cross-site interoperability, rather than say elaborate logical inferencing systems using ontologies. Yet the bewildering landscape of technology choices for using RDF in data platforms means that the developers suffer and still face a steep learning curve. For clients consuming Linked Data from multiple sources—*Linked Data Mashup* [Tran 2014]—the situation is still baffling in that relatively small differences in identifiers, vocabularies and usage patterns across deployment result in incompatibilities that may require platform-specific workarounds and mappings [Millard 2010].

The ecosystem of FAIR tooling is not currently mature enough to support Linked Data consumption in a user-friendly and efficient way [Thompson 2020], although recent metrics and tools for assessing *FAIRness* [Wilkinson 2018] can assist both data providers and consumers.

Evaluations by EOSC has since found that FAIRness metrics can vary widely across the different assessment tools for the same data resource [Wilkinson 2022a], showing that further definitions of conventions and practices are needed for consistent Linked Data publishing and consumption.

Making the FAIR principles achieve practical benefits for researchers and platform developers thus requires more specific constraints and broader consistency.

6.1.1.2 FDOs as a distributed object system on the Web

The framework-based comparisons in Section 3.1.3 on page 33 considered the implementation details of both FDO and Linked Data, and evaluated to what extent either can be considered a global distributed object system. The findings from this research show that FDO recommendations can benefit FAIR thinking to build machine actionable ecosystems and provide stronger promises of consistency and predictability across data platforms.

These comparisons highlighted that the Web on the other hand has a flexible, scalable and mature technology stack, which can form a solid basis for implementing FDO. However, if such implementation is to use Linked Data technologies, these must be constrained sufficiently in order to practically realize such an ecosystem within the FDO guidelines and without degrading the developer experience.

6.1.1.3 FDOs can be implemented on the Web using Signposting

Section 3.2.2 on page 72 explored how the FDO principles can be achieved for Linked Data as further constraints on existing standards. As Chapter 3 has highlighted throughout, there are many technical details remaining to be specified for FDO it to be consistently implemented according to its own principles.

If such conventions need to be evolved and specified no matter the protocol basis for FDO, this chapter argued, then it would be intuitive to build FDO on the mature Web stack, unless there was an compelling argument for alternative protocol stacks having other advantages.[1]

Section 3.2.3 found that the basis of Web-based FDOs can be built using only Signposting [Van de Sompel 2015, Van de Sompel 2022], adding a couple of non-intrusive HTTP headers that are agnostic to metadata standards and serializations. An implementation of such Web-based FDOs was shown in Section 4.2.

The Signposting approach has also been highlighted both by EOSC [Wilkinson 2022a, Wilkinson 2024] and as a possible FDO configuration type [Lannom 2022a]. The FAIR-IMPACT project launched an open call[2] where 14 participating institutions participated to build support for Signposting [Soiland-Reyes 2023b] in their data repositories and platforms. The results showed that development of "Webby FDOs" using Signposting for FDO structure and RO-Crate

[1] For instance, a de-centralised, resiliant architecture and long term preservation was the motivation for the design of the Interplanetary File System (IPFS) as a *Decentralized Web* [Trautwein 2022].

[2] https://fair-impact.eu/1st-open-call-support-closed

for metadata was largely achievable across participants with often little former experience, for a modest effort equivalent to 5 working days [Soiland-Reyes 2024c] or during a 5 day hackathon [Soiland-Reyes 2024a].

6.1.2 RO-Crate as a developer-friendly approach

As pointed out in Section 2.2.2 on page 26, while Linked Data is a powerful and flexible approach to publishing structured data on the Web, the developer experience of using Semantic Web technology still needs simplifications, like reducing number of choices for vocabularies and serialization formats.

Chapter 4 on page 75 introduced *RO-Crate* as a practical implementation of the FAIR principles for the purpose of packaging data alongside structured metadata. The approach builds on best practices for Linked Data, however RO-Crate specifications are example-driven with simple interrelated JSON structures, and primarily use a single, general purpose vocabulary.

This way of "Linked Data by stealth" means that developers don't need to be concerned about RDF implementation details, although they can at their option take advantage of RDF knowledge graph technologies like SPARQL (Section 4.1.2.2 on page 80). Extension points are well defined, and although extending RO-Crate do require some RDF knowledge like defining namespaces, reasonable examples and vocabulary repositories are provided by RO-Crate—developers do not for instance need to learn about ontologies nor need to deploy a web service serving multiple RDF serialisations for every described entity.

6.1.2.1 Just enough Linked Data

An important lesson from this work then is to use "just enough" Linked Data for the desired level of interoperability and knowledge representation. While previous efforts to 'FAIRify' largely have been concerned about representing the data values using an RDF data model, this can lead to significant effort needed in developing ontologies and vocabularies.

RO-Crate is using schema.org as its base vocabulary, and tries to follow its philosophy of building a lightweight semantic structure by associating many free text attributes to the same node, rather than making elaborate interconnected semantic objects. For instance, while a `Person`'s `affiliation` ideally goes to a `Organization` with it's own URL and other attributes, in some cases, a free text string is all information available, and this can be used cirectly as the `affiliation`.

With retrospect we can say that this reduction in semantic rigidity compared to use in OWL ontologies is a move back to the simplicity of early RDF as an open-ended model (see Section 2.2.1 on page 23), where a property can be used to point to almost anything, and RDF authors are free to use almost any term.

Another aspect that is not highlighted well in ontologies is where to stop in the formal knowledge representation of an object. In schema.org, many properties like http://schema.org/license are defined as having the *range*[3] of either `CreativeWork`[4] or `URL`[5], hinting that the licence is not required to be explained as another entity with further properties, but that the attribute's primary purpose is navigation or identification.

This would be a key aspect of Linked Data, which traditionally have had the rather undefined convention of "follow your nose" navigation—a client may attempt to request any node identifier (if it is a URL), and if, with content-negotiation, it returns some RDF, then that could be integrated into a joint knowlege graph, hopefully adding more description of that node, although possibly using other vocabularies. Signposting on the URL helps to make such navigation and expected profiles explicit.

However, ontologies used in Linked Data have not commonly indicated navigation waypoints as done in Schema.org, simply defining a property's range as a given class leaves it undefined if documents were expected to explain that node or link to its explanation. One notable exception is the Data Catalog Vocabulary (DCAT)[6] [Albertoni 2020] which have navigational properties like `dcat:landingPage` (to a `foaf:Document`) and `dcat:downloadURL` (to any `rdfs:Resource`).

6.1.2.2 Embedding contextual information reduces need for navigation

A divergence from common Linked Data practices is that our RO-Crate approach is making a self-described container. Rather than assume that information will always be available from the referenced URIs, and requiring clients to crawl their way through the many identifiers to see which ones contain more information, the RO-Crate contains a minimal description of each referenced contextual entity (Section 4.1.2.2 on page 82).

This has multiple purposes:

- Simplify user interfaces, e.g. show a human-readable label and type before the user chooses to click the link.
- Vocabulary adaptation, for instance describing with schema.org in the crate, what was expressed in FOAF vocabulary at the URI.
- Unify descriptions of semantic artefacts and web pages. Making "ad-hoc" semantic artefacts within the crate where none existed beforehand
- Embed "as of at time of writing" descriptions for longevity. An RO-Crate is self-contained and can be archived independently, and embedding contextual information reduces cross-organizational service dependencies (at the risk of outdated information).

[3] Expected type of object [Guha 2014], however note that schema.org uses http://schema.org/rangeIncludes instead of `rdfs:range`, to permit multiple alternatives without the need for a union class.

[4] http://schema.org/CreativeWork

[5] http://schema.org/URL

[6] Although the http://schema.org/Dataset type used by RO-Crate's root entity is derived from DCAT, RO-Crate does not assume a corresponding data catalogue on the Web.

Several of these reasons are also organizational in nature, reflecting back on the EOSC Interoperability Framework (Section 3.1.3.6 on page 63)—rather than requiring for instance the Research Organization Registry (ROR)[7] to add Linked Data representations of organizations, one can be made ad-hoc by the RO-Crate's author, and contained by the crate as a contextual entity.

This ability to describe a referenced entity locally is also a workaround for the chicken-and-egg problem of creating and linking Linked Data resources that vocabulary-wise are cross-related both ways. For instance orcid.org recently added schema.org content-negotiation, but *after* RO-Crate started describing people using the http://schema.org/Person type and ORCID identifiers.[8]

6.1.2.3 FDO ecosystems need to permit flexible references

When reflecting on the above contextualization from the propositions of FAIR Digital Objects as covered back in Chapter 3, we can predict a problem if every reference from an FDO must go to another pre-existing FDO (or at least a registered PID), in that there must then be a linear order of FDO creation within an ecosystem of compatible FDO types. A strict reading of the FDO principles means implementations cannot utilise the established human-readable Web for bootstrapping. This risks large cross-organizational delays with a stronger need for collaboration and coordination, or alternatively, starting with a smaller FDO data models that can gradually evolve to add more navigation, when and if registries appear with FDO interfaces.

The emphasis on strong typing in FDOs also means that seemingly incompatible types (for instance developed by the biodiversity community vs. those from genomics communities) lead to a split of the PID space of referenceable objects from a given type of FDOs. Counter to this, the current FDO recommendations for attributes and types [Blanchi 2023] do not require specification of the *range* of an attribute to be a PID of an FDO, and as current FDO type declarations have been relatively lightweight (textual descriptions only), they are flexible enough to permit URLs to any Web resource or existing Linked Data concepts.

There is a concern, however, that some FDO serializations using the Handle system and key-value attributes cannot distinguish between string literals and object references. Combined with the use of PID references expressed as handles rather than as a URIs (e.g. `21.14100/2fcf49d3-0608-3373-a47f-0e721b7eaa87` instead of https://hdl.handle.net/21.14100/2fcf49d3-0608-3373-a47f-0e721b7eaa87), this means that machine actionability suffers, in that the string value is not typed to what kind of reference it may or may not be, or in what PID system. Compare this with listing 2.1 on page 24 where the RDF syntax distinguishes literal strings from object references—in Handle-based FDOs, machine-actionable navigation is only possible by

[7] https://ror.org/

[8] One of my earlier code contributions to ORCID already established content-negotiation to RDF—but using the classical FOAF vocabulary [Brickley 2014]. Slightly inconsistent with Semantic Web principles[9], the registry is currently returning *Person* descriptions in different semantic models depending on the requested serialization. https://github.com/ORCID/ORCID-Source/blob/main/CONTENT_NEGOTIATION.md

[9] Signposting [Van de Sompel 2022] would indicate alternative vocabularies using distinct `profile` URIs.

understanding the attributes of the FDO type, yet as highlighted in the previous paragraph these type definitions are not directly machine-actionable themselves.

In schema.org we find a similar challenge with properties permitting both string values and object references. http://schema.org/keywords is perhaps the most ambiguous, as it permits `Text`[10], but also `URL` or `DefinedTerm`. The two latter cases are both intended for referencing controlled vocabularies, with the distinction that a `DefinedTerm` is defined explicitly within the referenced object, while the defined term is implied if only the URL is provided. JSON-LD contexts have the possibility of enforcing object references (`"@type: "@id"`), but this cannot be used in this case as freetext strings are also permitted. The result is that a freetext keyword that just looks like a URL cannot be distinguished from an intended URL reference, similar to the FDO Handle example in the previous paragraph.

In order to reduce such ambiguity and multiple developer choices, in RO-Crate all object references are in JSON-LD object form (as we saw in Listing 4.2 on page 87), and the RO-Crate context do not have any `@type` shortcuts for implicit references. RO-Crate 1.2 will also recommend that all entities[11] have a type, identifier and human-readable name.

6.1.2.4 Profiles restrict general flexibility to gain specific predictability

Section 4.1.4 on page 93 showed how RO-Crate is adopted by different scientific domains. Since the publication of the corresponding manuscripts in Chapter 4, RO-Crate has also been used by the Language Data Commons of Australia[12] [Smith 2022] building language corpa, Survey Ontology[13] [Scrocca 2021] describing surveys, DataPlant[14] for plant experiments, distributed provenance[15] of biological specimens [Wittner 2020, Wittner 2023a, Wittner 2023b], COVID-19 causal inferences[16] to compare public health interventions internationally [Meurisse 2023], and Trusted Research Environments[17] for controlled workflow computation on sensitive health data. Several of these use cases have also expanded RO-Crate with additional terms from schema.org or defined in corresponding RO-Crate profiles. The span of these domains shows that RO-Crate is flexible for a range of use cases and can be adopted by developers not familiar with Semantic Web technologies.

The discussion of strictness vs flexibility in Section 4.1.6.1 on page 106 highlighted the tension between a flexible open-ended model and the predictability needed to consistently create and consume content expressed by the model. While RO-Crate itself can be seen as a restriction of the more open-ended JSON-LD and schema.org, its extensibility points still allow different use

[10] To conflate matters, the `keywords` property can be repeated, but also allows multiple keywords within a single comma-separated string.

[11] https://www.researchobject.org/ro-crate/1.2-DRAFT/metadata.html#common-principles-for-ro-crate-entities

[12] https://www.researchobject.org/ro-crate/in-use/LDaCA.html

[13] https://www.researchobject.org/ro-crate/in-use/survey-ontology.html

[14] https://nfdi4plants.org/content/learn-more/annotated-research-context.html

[15] https://w3id.org/cpm/ro-crate

[16] https://w3id.org/ro/doi/10.5281/zenodo.6913045

[17] https://w3id.org/5s-crate/0.4

cases to expand on those conventions.

Section 4.1.2.4 on page 84 detailed how semi-formalised profiles can be gradually formed to at first *duck-type* a class of RO-Crates that have similar properties. Later work has formalised this as Profile Crate[18] to capture the profile itself as a separate crate. This have now evolved to use the W3C Profiles Vocabulary [Atkinson 2019] to explicitly link to vocabularies, mappings and importantly constraints expressed as RDF Shapes [Soiland-Reyes 2023c]. This turns RO-Crate profiles into machine-actionable type definitions, from which existing RDF tooling can do for instance validation. Figure 6.1 shows how the usage of *roles* within the profile crate indicates the purpose of the constituent parts. Roles are here particularly important as many of these Semantic Web resources are expressed in the same file format (e.g. `text/turtle`) and may be used for different purposes (e.g. SKOS is used to represent either a *mapping* or a *vocabulary* [Isaac 2009]).

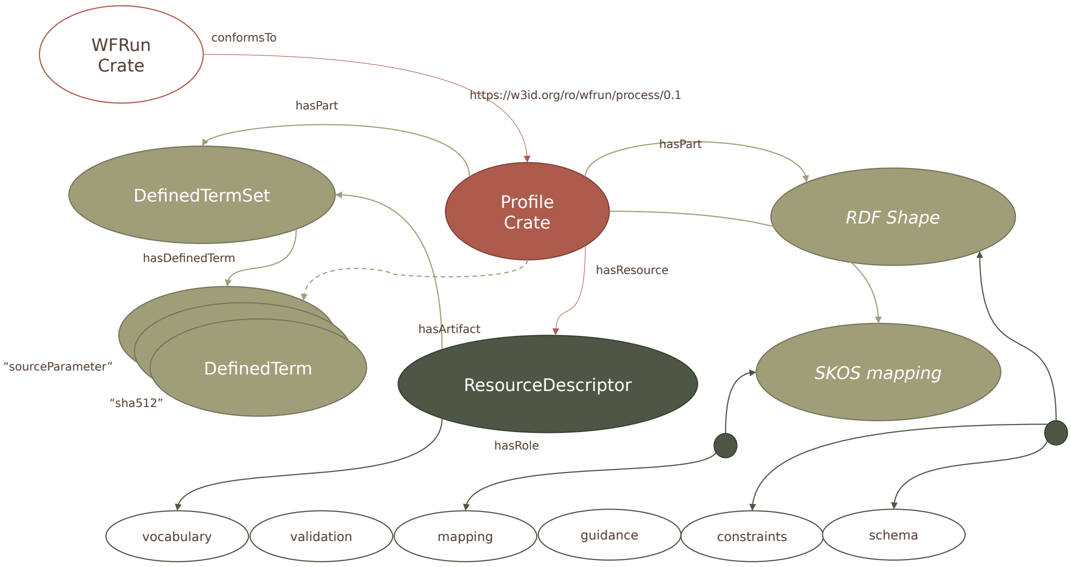


Figure 6.1: Example of Profile Crate for Workflow Run Crate. An RO-Crate *WFRun Crate* declares conformance with a given RO-Crate profile. Resolving the profile URI retrieves the *Profile Crate*, which parts include an *RDF Shape*, an *SKOS mapping* and a *DefinedTermSet*. By using the indirection of *ResourceDescriptor* from the Profiles Vocabulary [Atkinson 2019], the roles of each of these artefacts are defined, e.g. *constraints*. The embedded *vocabulary* as a *DefinedTermSet* defines ad-hoc terms like *sourceParameter* used by the Workflow Run Crate[19] profile [Leo 2024].

While profiles are at first lightweight indicators of common conventions for a class of crates (which may be implicit or explicit), they can be gradually formalised in a *eat own dogfood* way through another RO-Crate, optionally taking advantage of existing Semantic Web technology

[18] https://www.researchobject.org/ro-crate/1.2-DRAFT/profiles
[19] https://w3id.org/ro/wfrun/process/0.2

that enable for instance strict validation of domain-specific RO-Crates.

6.1.2.5 One vocabulary is not enough, but one profile may suffice

RO-Crate relies heavily on [schema.org] as its main vocabulary, but as highlighted in Section 4.1.7 on page 107 and 6.1.2.4 on page 191, domain-specific usage will eventually need to define their own terms in order to be specific enough for their use cases. However, we have found it is important to ensure a developer-friendly approach when specifying such profiles for RO-Crate—earlier work on ad-hoc terms[20] in RO-Crate used a simple CSV approach to be added to the ro-terms[21] namespace.

As with other aspects of RO-Crate, there is a gradual approach towards Linked Data practices. While conventional wisdom in Semantic Web would be to sit down and make your own ontology following design patterns [Blomquist 2009, Poveda 2010] and best practices for deployment [Matentzoglu 2022], in RO-Crate philosophy that would be more of a last resort. The middle of the ground is therefore adding the ad-hoc vocabulary directly to the profile crate, as shown in Figure 6.1 on the preceding page. In this approach a single profile URI can, through Linked Data and Signposting, play the role of:

- Human-readable documentation of conventions (negotiated to HTML preview).
- List of software and repositories the profile is intended for.
- List of additional schema.org types and properties utillised by the profile.
- Indication of which content is expected in the crate (e.g. a Workflow).
- Validation of a manifest conforming to the profile.
- Vocabulary definitions of additional terms.
- JSON-LD context which namespaces the additional terms (as any JSON-LD document can also be a JSON-LD context [Sporny 2020]).

It should be reasonable to expect developers able to make RO-Crates with their own additional terms to also be able to make a lightweight Profile Crate once those terms have stabilised. Developers with deeper familiarity with Semantic Web technologies can expand the profile capabilities to use existing ontology methodologies, in which case it would be preferrable to aggregate separate semantic artefacts from the Profile Crate rather than embedding them in the RO-Crate Metadata File.

In the FAIR-IMPACT project we are evaluating if the Profile Crate approach is also suitable for FAIR publishing of semantic artefacts themselves, e.g. ontologies and mappings. This is an attractive proposal because such artefacts are also becoming multifaceted, with multiple formats and profiles (e.g. an ontology expressed with OWL2 RL in RDF Turtle syntax), documentation and similar attribution and provenance challenges which RO-Crate is built to handle.

[20] https://www.researchobject.org/ro-crate/1.1/appendix/jsonld.html#adding-new-or-ad-hoc-vocabulary-terms
[21] https://github.com/ResearchObject/ro-terms

6.1.3 Future RO-Crate directions

In this section we consider future directions for RO-Crate and ongoing RO-Crate adaptations not covered by Section 4.1.

6.1.3.1 User applications are needed for researchers to generate FAIR Research Objects

RO-Crate and its best practices can be considered a type of *middleware* used by application developers to capture and transmit metadata and relate data files that together form some tangible unit (a *Research Object* (*RO*) [Bechhofer 2013]). While RO-Crate have already been implemented by several repositories and applications such as workflow systems, it is important to also consider the role of user applications in order to increase adoption of FAIR Research Objects by scholars in general.

Template-based crates with ya2ro Futher work by the RO-Crate community has created more user-fronting tools such as Pavel 2023,[22] which given metadata and identifier in a YAML file can retrieve contextual metadata from ORCID, GitHub and DOI registries and build and publish a completed RO-Crate [Pavel 2023]. While this technology still requires some understanding of editing, it is intended to be more approachable to data scientists and for use with simple Web publishing platforms like GitHub Pages.[23] The GitHub Action ro-crate-preview[24] also automate HTML preview generation of crates on GitHub Pages.

Editing and publishing RO-Crate in ROHub The repository ROHub[25] [Garcia-Silva 2019] has recently added RO-Crate import and export [Fouilloux 2023], and provides both a browseable repository for publishing crates, but also interactive and collaborative editing of its metadata. In this use case, RO-Crate plays the role as an exchange and archiving format, as the hub stores the crates in general-purpose repositories Zenodo and B2Share which do not have the facility to keep the granular metadata expressed within the RO-Crate metadata file. As detailed in [Fouilloux 2023], a series of templates assist users in creating research objects with particular content and annotations.

Making ad-hoc vocabularies in Crate-O The Crate-O[26] tool has been developed by Language Data Commons of Australia (LDaCA)[27] as a general-purpose RO-Crate editor and successor to Describo [La Rosa 2021d] and Describo Online [La Rosa 2021c]. This tool can describe any folder and resources from the Web as an RO-Crate, supporting any schema.org type and property, pluggable with any RDFS vocabularies [Guha 2014]. Notably this tool is also intended for

[22] https://github.com/oeg-upm/ya2ro
[23] https://pages.github.com/
[24] https://github.com/marketplace/actions/ro-crate-preview
[25] https://www.rohub.org/
[26] https://language-research-technology.github.io/crate-o/
[27] https://ldaca.edu.au/

creation of such vocabularies, and is thus a lightweight user interface for building a Profile Crate (Section 6.1.2.4 on page 191) using Schema.org style Schemas[28] (SoSS)[29].

Executable papers can fully represent their computation using RO-Crate LivePublication[30] [Ellerm 2023] is a proof of concept of an *executable paper*, which interactive visualization and statistical calculations can be regenerated on the fly taking into consideration data sources updated after the paper's publication date. A corresponding RO-Crate[31] is the mechanism to enable this execution on the Globus infrastructure through an innovative use of individual RO-Crates and containers for each computable element of the paper, nested within a top-level Crate for the paper.

This novel approach shows how it is possible to use RO-Crate as an machine-actionable object, which do not rely on bundling an underlying workflow representation in an existing workflow language.

6.1.3.2 Web-based FDOs can use RO-Crate for its metadata

Section 4.2 on page 109 argues that many of the FDO requirements [Anders 2023a] for metadata can be implemented as *RO-Crate FDOs* (Section 5.2.2.3 on page 139), with FAIR Signposting [Van de Sompel 2015, Van de Sompel 2022] assisting navigation from persistent identifiers, and the RO-Crate containing the metadata.

This approach was first implemented in the repository WorkflowHub [Goble 2021, Wittenburg 2022a], and in the FDO Forum [Van de Sompel 2023] suggests RO-Crate with Signposting as a modern update to his OAI-ORE[32] approach from 2008 [Lagoze 2008]. RO-Crate FDOs are being further developed within the Horizon Europe projects EuroScienceGateway[33] [Soiland-Reyes 2022f] and FAIR-IMPACT[34] [Goble 2022].

RO-Crate FDOs complements the findings of Section 6.1.1.3 on page 187, in that RO-Crate provides FDO with a generic metadata framework and a serialization that can work both for FDOs on the Web and with legacy Handle/DOIP approaches—this metadata role for RO-Crate in the FDO ecosystem is also highlighted by [Wittenburg 2023b].

Some extra considerations is rightly needed on identifiers to reduce relative paths challenges with RO-Crate FDOs—for this purpose, the next specification [RO-Crate 1.2] introduce a distinction

[28] It is notable that schema.org's own vocabulary definition use RDFS directly for Linked Data interoperability, rather than its own http://schema.org/Class, http://schema.org/Property, or the SKOS-like http://schema.org/DefinedTerm. On property definitions, SoSS use http://schema.org/domainIncludes and http://schema.org/rangeIncludes to avoid union classes for alternative domain/range types, which can clutter OWL/RDFS equivalent properties.

[29] https://schema.org/docs/schemas.html

[30] https://livepublication.github.io/LP_Pub_LID/

[31] https://livepublication.github.io/LP_Pub_OrchestrationCrate/

[32] OAI-ORE was also used by earlier Research Object approaches [Belhajjame 2015, Soiland-Reyes 2014] to capture the aggregation aspect of ROs.

[33] https://eurosciencegateway.eu/

[34] https://fair-impact.eu/

between an attached RO-Crate[35] (*has some root directory which may contain other files referenced by relative paths, possibly archived in a ZIP or exposed on the Web*) and a detached RO-Crate[36] (*no defined root directory, all references are absolute*). Although both style of crates can contain absolute URI references, this *detached* style is more suitable for an FDO architecture, even for use within APIs which do not lend themselves to relative path references (such as DOIP-over-HTTP [CNRI 2023a]). RO-Crate 1.2 also define methods for converting between attached/detached[37] crates using standard JSON-LD tooling, showing another advantage of using Linked Data as basis for RO-Crate.

6.1.3.3 How FAIR are RO-Crates?

FAIROs [González 2022] is a framework for calculating a "FAIRness" score for research objects. For RO-Crate evaluation this puts additional requirements on the use of persistent identifier for the RO-Crate, and that the core metadata of the crate (e.g. licensing) is provided. These aspects are important for ensuring FDO machine actionability of RO-Crates.

Another aspect of FAIRness for RO-Crate is if extensions are themselves following FAIR principles (RDA-I2-01M *Metadata uses FAIR-compliant vocabularies*). The Profile Crate specifications for extension vocabularies[38] recommend the use of `DefinedTerm` or `DefinedTermSet` as a mechanism to "import" an existing term or vocabulary to the profile, allowing a neutral way to define these terms independent of their ontology technology. There is some tension with Crate-O's "Schema.org style Schemas" (see Section 6.1.3.1 on page 194) which desired compatibility with `rdfs:Class` and `rdfs:Property` and wide-spread RDFS tooling—the RO-Crate community consensus is to use the rdfs types when the term is *defined* by the profile rather than imported and to avoid the RDFS-like http://schema.org/Class and http://schema.org/Property overall.

The use of profiles, and particularly nested profiles as in Figure 5.11 on page 164, makes validation of RO-Crates more complex. An initial approach of ShEx validation in runcrate[39] extended the ro-crate-validator-py[40] library to use ShEx based validation [Baker 2019] depending on declared WRROC profiles. Future work[41] is planned to further investigate this using a combination of Semantic Web and RDF Shapes technologies, possibly using hierarchical profile validation with Cheka[42] but based on the crate's declared `conformsTo` statements.

6.1.3.4 RO-Crate can build collections of digital objects

RO-Crate has also been proposed as a generic mechanism for FDO Collections [Soiland-Reyes 2023c], as an aggregator of FDOs by their PIDs. Such collections add a similar challenge in FDO as in

[35] https://www.researchobject.org/ro-crate/1.2-DRAFT/structure.html#attached-ro-crate
[36] https://www.researchobject.org/ro-crate/1.2-DRAFT/structure.html#detached-ro-crate
[37] https://www.researchobject.org/ro-crate/1.2-DRAFT/appendix/relative-uris.html
[38] https://www.researchobject.org/ro-crate/1.2-DRAFT/profiles#extension-vocabularies
[39] https://github.com/ResearchObject/runcrate/pull/17
[40] https://github.com/ResearchObject/ro-crate-validator-py
[41] https://s11.no/2023/comp66090/profiles/
[42] https://github.com/surroundaustralia/cheka

Linked Data, in that clients may need to resolve an excessive number of persistent identifiers (see Section 2.2.1 on page 26) to FDOs which may be of different semantic types. Using a detached RO-Crate for such collections, the bibliographic metadata of each PID can be directly embedded and normalised to a single vocabulary, reducing client needs for recursive queries and type mappings.

Work on building large data citations as a "reliquary"—a *container of persistent identifiers* (*PIDs*) [Buck 2022]—started from the earth science domain with AGU's Data Citation Community of Practice[43] and continues in RDA's Complex Citation Working Group[44]. In this approach RO-Crate is being considered as a promising implementation to capture large number of citations along with minimal metadata, including licensing and attribution. Here a main motivation is to avoid excessive lists of data citations for scholarly publications following processing of aggregated datasets from repositories such as GBIF [GBIF 2021], while still propagating each dataset's FAIR metadata (as required by the Creative Commons Attribution licence) through the indirection of a collection. There is a potential overlap with workflow run provenance, although a workflow is not required by reliquaries.

6.1.3.5 Mutable FDOs can be captured in knowledge graphs using RO-Crate

Knowledge Enhanced Digital Objects (KEOD)[45] [Luo 2022] is an experimental approach of building a data lake using a combination of knowledge graphs, RO-Crate and PID records [Luo 2023]. This is effectively an FDO implementation: A KEDO PID is a Handle that identifies a KEDO Object, described using a KEDO RO-Crate. This crate again has *internal RO-Crate*s as parts, which records a combination of *Features* and *Insights*. The distinction is that features are mainly fixed at digital object creation and considered directly describing the object's nature, while insights can be discovered later from further processing and linkage. This approach solves a mutability problem in FDOs, as the KEOD system only allows insights to be added along with provenance that connect PIDs when KEDOs evolve. Files in a KEDO RO-Crate are stored locally, and each recorded with a Handle PID within the crate.

This KEOD setup of multiple graphs forming a single knowledge unit can be considered analogu-ous to nanopublications [Kuhn 2021] but for FDOs. Indeed using nanopublication to capture FDOs of digital twins has also been proposed [Schultes 2022], however, that use a different distributed architecture where the PIDs for nanopublications are generated by cryptographically hashing their content [Kuhn 2021].

[43] https://data.agu.org/DataCitationCoP/

[44] https://www.rd-alliance.org/groups/complex-citations-working-group

[45] https://github.com/luoyu357/KEDODataLake

6.1.3.6 Distributed architectures for FAIR Digital Objects can use detached crates

The DeSci Nodes[46] system has been developed by the DeSci foundation[47], where dPID[48] (distibuted Persistent Identifier) act as an overlay of the Interplanetary File System (IPFS) [Trautwein 2022]. Users can interact with the DeSci platform for building and publishing Research Objects, and the DeSci metadata[49] are exposed as a detached RO-Crate[50] with IPFS references (see example dPID[51]). DeSci Nodes have documented a FAIR Implementation Profile[52] (FIP) [Schultes 2020] documenting compliance with FAIR principles.

This is a novel FAIR Digital Object implementation that challenges both the traditional centralised FDO approach using the Handle system, as well as the mostly Web-based RO-Crate ecosystem covered in Section 4.1 on page 77. It remains to be independently verified if the decentralisation of IPFS is effectively constrained by access through a centralised API, or if dPIDs can be retrieved from multiple independent resolvers.

The use of detached crates has also been utilised by the Language Data Commons of Australia Program[53], where RO-Crate is part of navigating centralised API resources, rather than a standalone publication on the Web. In both of these approaches, additional FDO measures such as using persistent identifiers and validation against profiles become important.

6.1.4 Workflows capture computational methods

Chapter 5 on page 119 explored in depth different ways in which FAIR Digital Objects and RO-Crate are applied to computational workflows, in effect capturing the computational methods in a FAIR Research Object.

6.1.4.1 Workflows can be constructed of FAIR digital objects

As introduced in Section 1.2.3 on page 10, we have previously proposed the concept of *FAIR Computational Workflows* [Goble 2020]. That work expands on the well-established motivations for using scientific workflows systems [Möller 2017, Atkinson 2017], such as supporting Automation, Scalable execution, Abstraction, and Provenance [Ludäscher 2016], and highlights that workflows themselves benefit from and contribute to FAIR data, for instance providing metadata for describing workflow outputs. In addition workflow themselves can be considered digital objects that should be shared as a reproducible computational method.

Applying the FAIR principles for workflows in practice has however revealed additional challenges, such as lack of clarify of what constitutes a *workflow* as opposed to FAIR Research

[46] https://docs.desci.com/

[47] https://www.descifoundation.org/

[48] https://www.dpid.org/

[49] https://docs.desci.com/learn/open-state-repository/metadata

[50] https://www.researchobject.org/ro-crate/1.2-DRAFT/structure.html#detached-ro-crate

[51] https://beta.dpid.org/46?jsonld

[52] https://docs.desci.com/learn/fair-data/fair-compliance/desci-nodes-fip

[53] https://www.researchobject.org/ro-crate/in-use/LDaCA.html

Software in general [Katz 2021b], or reduced reusability when the workflow requires unwritten, human-centric operations between computational steps (e.g. trivial file column manipulations) [Wilkinson 2022b].

In developing the repository WorkflowHub[54] [Goble 2021] we emphasised the importance of preserving and publishing not just executable workflow definitions, but their structured descriptions independent of workflow formats as well as further references to external sources, software required (including the workflow engine). For this we developed the Workflow RO-Crate Profile [Bacall 2022], which became a foundational format for more specific workflow execution profiles (Section 5.4 on page 152).

One aspect that makes workflow management systems different from Research Software in general, is that they frequently encourage modularization, in that the composition of steps also can reflect the analytical process that is intended by the scientists. Mature workflow systems like Galaxy [Galaxy 2022] provide a large collection of re-usable components that wrap underlying command line tools and make them interoperable without manual adjustments. In CWL [Crusoe 2022], tool definitions include not just execution details, but also structured input/output definitions, allowing them to be reused and combined in multiple workflows.

Section 5.1 on page 121 explored how such building blocks can themselves be considered FAIR digital objects. These assist workflow systems in propagating rich metadata about tools and their analytical purpose, but also allows building blocks to be reused across workflow systems. This in effect means that a *canonical workflow Wittenburg 2022b* can be implemented in different workflow languages, each executing the same *canonical steps* in the same way. Given that FAIR Digital Objects emphasize machine-actionability, and we can consider workflows as FDOs, it is important to have the ability not just to reliably re-execute a workflow, but even re-use its constituent steps.

6.1.4.2 Building FDOs incrementally challenges typing constraints

Sections 5.2 on page 133 and 5.3 on page 148 showed another aspect of using such building blocks, where FDOs are the unit of communication between steps in the workflow. This approach is pushing the envelope of FAIR Digital Objects concept, by having the FDO built incrementally by different stages of the specimen digitization pipeline, and exchanged only within the workflow system before it is ready to be published. This scenario, where we experimented with strict validation with JSON Schema for every step, highlighted limitations of having larger composite FDO object types, as intermediate FDOs would not validate without significantly softening the schema. In effect the Specimen Data Refinery (SDR) workflow can be considered as a variant of the classic *Builder pattern* [Gamma 1995, pp. 97–106], which gradually constructs an object through a series of operations on an intermediate object (the builder). However, as highlighted by Section 5.3.2 on page 150, ducktyping-like profiles/interfaces are needed for typing incremental FDOs, to ensure that a later workflow step receives a partial FDO with the

[54] https://workflowhub.eu/

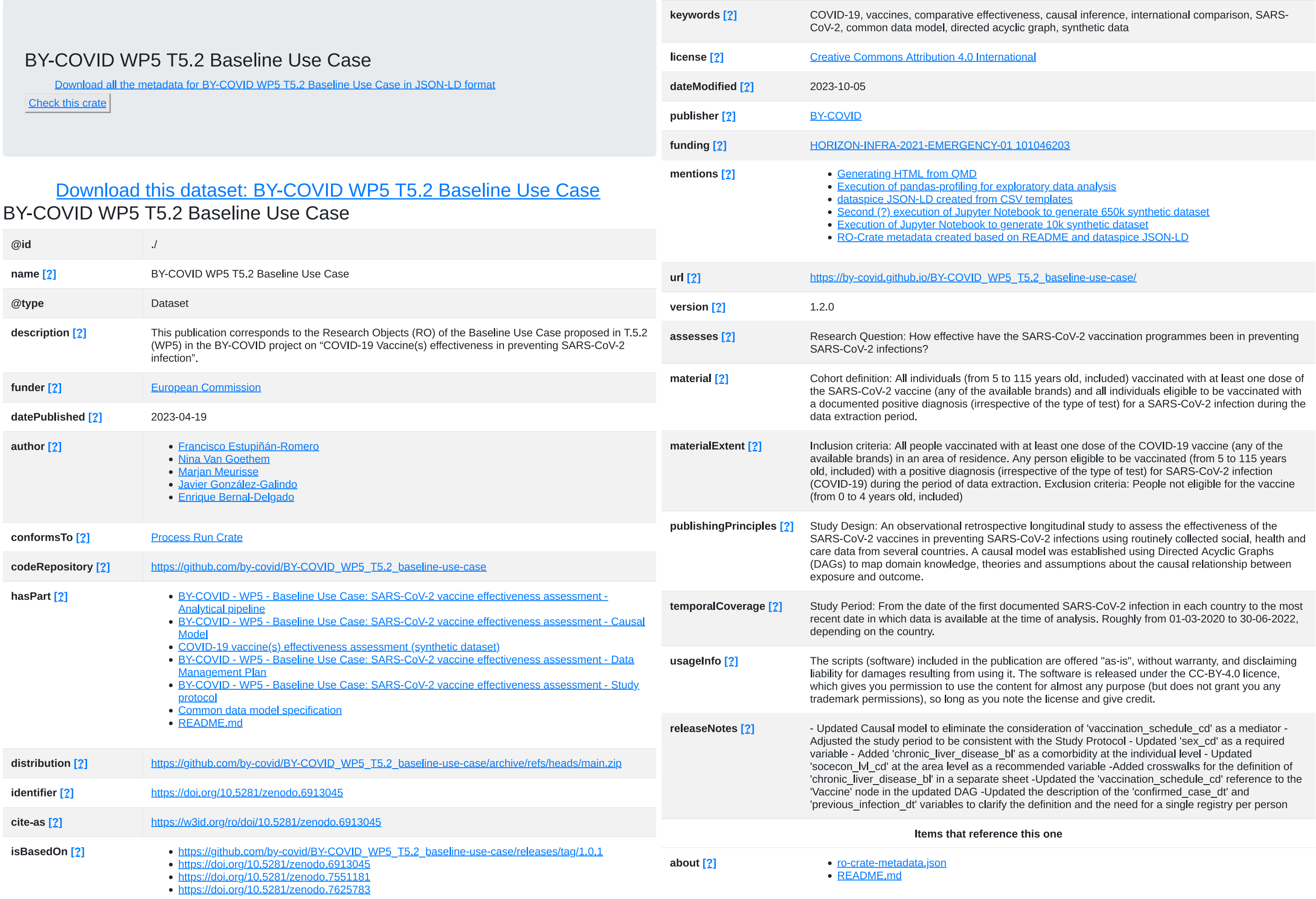
expected fragments populated, otherwise workflow users would be able to compose steps with the FDO in an invalid state.

The later Section 5.4.2 on page 155 showed how RO-Crate profiles for workflow provenance FDOs can be staggered for different granularity levels, but that is more akin to a class hierarchy, as each level builds on previous complete levels. The Five Safes Crate profile (Section 5.4.5.3 on page 179) however, has a similar incremental pattern as the SDR FDOs, and the different review states should be performed in a particular order. Enforcing this for typing purposes may require explicit rule-based abstract state machines (ASM) [Gurevich 1995], as has been demonstrated for Linked Data with ASM4LD [Käfer 2018a, Käfer 2018b].

6.1.4.3 Flexible profiles increase adaptability of interoperable provenance

Section 5.4 on page 152 introduced the Workflow Run RO-Crate (WRROC) profiles for capturing workflow provenance. It was highlighted that the multiple levels were designed to ease adoptability—indeed the different WRROC implementations (Section 5.4.3 on page 162) have chosen profiles depending on the provenance available to that particular engine. In addition, some implementations had to utilise optionality of some attributes, for instance to handle dynamic workflows.

In addition the *Process Run Crate* profile was shown to be suitable also for "manual workflows" where processes are executed by hand, as illustrated by Figure 6.2 on the next page. In this example, the process run is only a small part of the crate, namely to generate the synthetic dataset, but of bigger importance in this crate is the causal model that explains to humans the relationships that led to the synthetic dataset. There is no overall computational workflow as the individual computational steps are performed with human interaction; however, this also means the RO-Crate metadata must be created by human interaction.



BY-COVID WP5 T5.2 Baseline Use Case

Download all the metadata for BY-COVID WP5 T5.2 Baseline Use Case in JSON-LD format

Check this crate

Download this dataset: BY-COVID WP5 T5.2 Baseline Use Case

BY-COVID WP5 T5.2 Baseline Use Case

@id	./
name [?]	BY-COVID WP5 T5.2 Baseline Use Case
@type	Dataset
description [?]	This publication corresponds to the Research Objects (RO) of the Baseline Use Case proposed in T.5.2 (WP5) in the BY-COVID project on “COVID-19 Vaccine(s) effectiveness in preventing SARS-CoV-2 infection”.
funder [?]	European Commission
datePublished [?]	2023-04-19
author [?]	• Francisco Estupiñán-Romero • Nina Van Goethem • Marjan Meurisse • Javier González-Galindo • Enrique Bernal-Delgado
conformsTo [?]	Process Run Crate
codeRepository [?]	https://github.com/by-covid/BY-COVID_WP5_T5.2_baseline-use-case
hasPart [?]	• BY-COVID - WP5 - Baseline Use Case: SARS-CoV-2 vaccine effectiveness assessment - Analytical pipeline • BY-COVID - WP5 - Baseline Use Case: SARS-CoV-2 vaccine effectiveness assessment - Causal Model • COVID-19 vaccine(s) effectiveness assessment (synthetic dataset) • BY-COVID - WP5 - Baseline Use Case: SARS-CoV-2 vaccine effectiveness assessment - Data Management Plan • BY-COVID - WP5 - Baseline Use Case: SARS-CoV-2 vaccine effectiveness assessment - Study protocol • Common data model specification • README.md
distribution [?]	https://github.com/by-covid/BY-COVID_WP5_T5.2_baseline-use-case/archive/refs/heads/main.zip
identifier [?]	https://doi.org/10.5281/zenodo.6913045
cite-as [?]	https://w3id.org/ro/doi/10.5281/zenodo.6913045
isBasedOn [?]	• https://github.com/by-covid/BY-COVID_WP5_T5.2_baseline-use-case/releases/tag/1.0.1 • https://doi.org/10.5281/zenodo.6913045 • https://doi.org/10.5281/zenodo.7551181 • https://doi.org/10.5281/zenodo.7625783
keywords [?]	COVID-19, vaccines, comparative effectiveness, causal inference, international comparison, SARS-CoV-2, common data model, directed acyclic graph, synthetic data
license [?]	Creative Commons Attribution 4.0 International
dateModified [?]	2023-10-05
publisher [?]	BY-COVID
funding [?]	HORIZON-INFRA-2021-EMERGENCY-01 101046203
mentions [?]	• Generating HTML from QMD • Execution of pandas-profiling for exploratory data analysis • dataspice JSON-LD created from CSV templates • Second (?) execution of Jupyter Notebook to generate 650k synthetic dataset • Execution of Jupyter Notebook to generate 10k synthetic dataset • RO-Crate metadata created based on README and dataspice JSON-LD
url [?]	https://by-covid.github.io/BY-COVID_WP5_T5.2_baseline-use-case/
version [?]	1.2.0
assesses [?]	Research Question: How effective have the SARS-CoV-2 vaccination programmes been in preventing SARS-CoV-2 infections?
material [?]	Cohort definition: All individuals (from 5 to 115 years old, included) vaccinated with at least one dose of the SARS-CoV-2 vaccine (any of the available brands) and all individuals eligible to be vaccinated with a documented positive diagnosis (irrespective of the type of test) for a SARS-CoV-2 infection during the data extraction period.
materialExtent [?]	Inclusion criteria: All people vaccinated with at least one dose of the COVID-19 vaccine (any of the available brands) in an area of residence. Any person eligible to be vaccinated (from 5 to 115 years old, included) with a positive diagnosis (irrespective of the type of test) for SARS-CoV-2 infection (COVID-19) during the period of data extraction. Exclusion criteria: People not eligible for the vaccine (from 0 to 4 years old, included)
publishingPrinciples [?]	Study Design: An observational retrospective longitudinal study to assess the effectiveness of the SARS-CoV-2 vaccines in preventing SARS-CoV-2 infections using routinely collected social, health and care data from several countries. A causal model was established using Directed Acyclic Graphs (DAGs) to map domain knowledge, theories and assumptions about the causal relationship between exposure and outcome.
temporalCoverage [?]	Study Period: From the date of the first documented SARS-CoV-2 infection in each country to the most recent date in which data is available at the time of analysis. Roughly from 01-03-2020 to 30-06-2022, depending on the country.
usageInfo [?]	The scripts (software) included in the publication are offered "as-is", without warranty, and disclaiming liability for damages resulting from using it. The software is released under the CC-BY-4.0 licence, which gives you permission to use the content for almost any purpose (but does not grant you any trademark permissions), so long as you note the license and give credit.
releaseNotes [?]	- Updated Causal model to eliminate the consideration of 'vaccination_schedule_cd' as a mediator - Adjusted the study period to be consistent with the Study Protocol - Updated 'sex_cd' as a required variable - Added 'chronic_liver_disease_bl' as a comorbidity at the individual level - Updated 'socecon_lvl_cd' at the area level as a recommended variable -Added crosswalks for the definition of 'chronic_liver_disease_bl' in a separate sheet -Updated the 'vaccination_schedule_cd' reference to the 'Vaccine' node in the updated DAG -Updated the description of the 'confirmed_case_dt' and 'previous_infection_dt' variables to clarify the definition and the need for a single registry per person

Items that reference this one

about [?]	• ro-crate-metadata.json • README.md

Figure 6.2: Example of RO-Crate using the Process Run Crate profile to describe a BY-COVID use case for modelling vaccine effectiveness [Meurisse 2023]. The crate *hasPart* multiple data entities, and *mentions* several process runs according to the profile. The use case is further described with extra schema.org attributes like *temporalCoverage*. Screenshot of RO-Crate preview HTML, modified for print from https://w3id.org/ro/doi/10.5281/zenodo.6913045

The community making WRROC involved multiple developers from different backgrounds and a variety of workflow systems. A lesson to learn from that experience is that even though RO-Crate profiles give additional rigidity that enable interoperability like the *runcrate run* reproducibility in Section 5.4.3.1 on page 164, profiles need to also ensure sufficient flexibility for individual implementations of different capabilities and purposes. Profiles are therefore different from stricter type systems and bounded schemas as otherwise used by FDO implementations and Linked Data ontologies.

An interoperability challenge remains on how much flexibility to permit, as discussed in 6.1.2.4 on page 191. However it is arguably more interoperable to have optional features defined by the community when needed, rather than individual vendor extensions—giving common ground and graceful fallback. This practice is in line with FDO principle FDO-FDOR4 (*can include other community defined and registered attributes*) [Anders 2023a] and FAIR principle RDA-R1.3-01M (*Metadata complies with a community standard*) [FAIR Maturity 2020].

6.1.4.4 Profiles should not need to define subclasses

Part of RO-Crate's philosophy is to use Semantic Web technology while avoiding many of its pitfalls discussed in Section 2.2.1 on page 23 and 4.1.2.3 on page 84. An unusual consequence of this is that extension profiles like WRROC end up reusing multiple types for the same entity. For instance, the *CreateAction* representing a process run[55] has an *instrument* to either a `SoftwareApplication`, `SoftwareCourceCode` or `ComputationalWorkflow`. In the specialising profile for workflow run[56] the actual type of the instrument is however a combination: `["File", "SoftwareSourceCode", "ComputationalWorkflow"]`

Traditional ontology thinking such as with OWL and RDFS would be to create a class hierarchy, indeed both http://schema.org/SoftwareSourceCode and http://schema.org/MediaObject (aliased as `File` in RO-Crate's JSON-LD) are subtypes of http://schema.org/CreativeWork which has most of the useful properties like *author* and *license*. It would however not be RO-Crate's job to modify the existing schema.org class hierarchy to inject artificial superclasses like *ApplicationOrSourceCodeOrWorkflow*, neither would it be desirable to create a subproperty of *instrument* with the intended *domainIncludes*, as that means using custom terms that diverge from schema.org.

It is an important part of the simplification of the Semantic Web in RO-Crate that consumers should not need to do any ontology retrieval or reasoning. For RO-Crate users it is therefore natural to occasionally combine types, enabling properties from both. schema.org is not a strict ontology with disjoint classes, so this usually do not cause any problem. It is however not desirable to re-iterate supertypes already defined within schema.org such as *CreativeWork*.

RO-Crate profiles like WRROC are doing a combination of *restrictions* (requiring the crate to have a particular entity) and *extensions* (suggesting additional terms to use). By following the

[55] https://w3id.org/ro/wfrun/process/0.4
[56] https://w3id.org/ro/wfrun/workflow/0.4

RO-Crate philosophy the profiles also reuse existing schema.org types as much as possible, adding a filtered overlay of their many properties, just like RO-Crate itself, and only adding terms where no appropiate alternative exist.

Listing 6.6 shows an attempt to declare the partial WRROC requirements from the top of this section in OWL. In doing so it was necessary to introduce additional types for the profile and the action, in addition to anonymous union classes and inverse properties. It is clear that expressing a full RO-Crate profile in this matter would require a deep understanding of OWL ontologies and would require its own set of unit tests, and would not be inline with the RO-Crate philosophy of *just enough Linked Data*.

```
<https://w3id.org/ro/wfrun/process/0.4>
  rdf:type owl:NamedIndividual, :ProcessRunCrateProfile ;
  rdfs:label "Process Run Crate profile 0.4"@en-gb .
:ProcessRunCrateProfile rdf:type owl:Class ;
    rdfs:subClassOf :ROCrateProfile ,
      [ rdf:type owl:Restriction ;
        owl:onProperty [ owl:inverseOf dct:conformsTo ] ;
        owl:allValuesFrom [
            rdf:type owl:Restriction ;
            owl:onProperty s:mentions ;
            owl:someValuesFrom :ProcessRunAction
          ]
      ] ;
    rdfs:label "Process Run Crate profile"@en-gb .
:ProcessRunAction rdf:type owl:Class ;
  owl:equivalentClass [
    rdf:type owl:Class;
    owl:intersectionOf (
      [ rdf:type owl:Class ;
        owl:unionOf ( s:ActivateAction s:CreateAction s:UpdateAction ) ]
      [ rdf:type owl:Restriction ;
        owl:onProperty s:instrument ;
        owl:someValuesFrom [ rdf:type owl:Class ;
            owl:unionOf ( s:SoftwareApplication s:SoftwareSourceCode
              bioschemas:ComptuationalWorkflow ) ]
      ] )
  ] ;
  rdfs:subClassOf s:Action ;
  rdfs:label "ProcessRunAction"@en-gb .
```

Listing 6.6: Defining a Process Run action as an OWL equivalence class. A versioned `:ProcessRunCrateProfile` is given a restriction in that the RO-Crate root which list the profile as `conformsTo` must *mention* an action (one of *ActivateAction*, *CreateAction*, *UpdateAction*) which *instrument* have at least one instance of *SoftwareApplication SoftwareSourceCode* or *ComptuationalWorkflow*. This OWL ontology uses equivalence classes as the WRROC types must be inferred and not declared in their JSON-LD `@id`. OWL Turtle snippet modified from https://github.com/ResearchObject/workflow-run-crate/pull/69

However, that is not to say that such OWL rules cannot be generated from simpler definitions of

RO-Crate profile requirements. Of future consideration is that the tool Crate-O's Editor profiles[57] (mainly for driving the UI form generations), and LinkML[58] which can generate ShEx, SHACL, OWL and JSON-LD context from a concise YAML definition—in coordination with validation as discussed in Section 6.1.3.3 on page 196.

6.1.4.5 Linked data provenance models can be made approachable

As discussed in Sections 1.2.3 on page 10 and 5.4.1 on page 152, the subject of capturing provenance from computational workflow executions is both diverse and mature, with most of the implementations coalescing on the W3C PROV data model [Moreau 2013], and in particular specializations of the OWL ontology serialization PROV-O [Lebo 2013a]. Yet even with earlier approaches like Research Objects wfprov [Belhajjame 2015], D-PROV [Missier 2013] and CWLProv [Khan 2019], the uptake of these technologies by Workflow Management Systems is fragmented at best.

Given these approaches rely on Semantic Web technology and hierarchies of ontologies, it is not a far stretch to hypothesise that some of the challenges on uptake of Linked Data we discussed in Section 2.2 on page 23 and 4.1.2.3 on page 84 also apply to the use of PROV-O by workflow systems developers.

A core concern for workflow users is to get hold of and distribute their result data—the exact computational structure is often of less concern as it is taken for given from the workflow definition. A change of focus from process-oriented to data-oriented will therefore also be beneficial for workflow provenance.

Targeted workflow provenance models like Biocompute Objects (BCO) [IEEE 2791-2020, Alterovitz 2018], discussed in Sections 4.1.4.2 on page 96 and 5.4.5.4 on page 182, emphasise the *context* of the workflow—what is the purpose? What are possible inputs? What data sources are referenced? BCO is being implemented by workflow management systems in Life Science domain including Nextflow and Galaxy, but is perhaps not deemed generic enough for other domains like earth sciences or astronomy. While extensions are possible in BCO (e.g. FHIR[59]) they are separate from the rest of the descriptions and do not natively support Linked Data principles.

The aspect of documenting the context and human processing is also emphasised by Five Safes Crate [Soiland-Reyes 2023e] as discussed in Section 5.4.5.3 on page 179. Here the [schema actions] are used to record the review process in a restricted environment for sensitive data, while also progressing the crate towards becoming a Workflow Run Crate. This model has built interest beyond workflow computations in the health data research area, amongst implementers who are not native speakers of FAIR principles or Linked Data technologies.

Section 5.4.3 on page 162 presented the range of workflow systems that have implemented

[57] https://github.com/Language-Research-Technology/ro-crate-editor-profiles

[58] https://linkml.io/linkml/

[59] https://wiki.biocomputeobject.org/index.php?title=Extension-fhir

WRROC, and Section 5.4.4 on page 171 illustrated usage in the biomedical domain. While it is too early to tell to what extent WRROC is an approachable lightweight provenance model that can be implemented for many different research domains, the reception from those approaching it so far has been overwhelmingly positive. At the same time, new members of the WRROC community tend to contribute new requirements or adjustments that means the model is both maturing and evolving.

6.1.4.6 Combining provenance and metadata models gives the best of both worlds

The intention of Workflow Run Crate, and indeed RO-Crate overall, is not to replace all existing Linked Data descriptions of research data and workflows. Even if the format of RO-Crate is JSON-LD, and in theory can support any RDF vocabulary, that does not mean that doing so is the right design decision. The engineering principle of *separation of concerns* applies just as well to Linked Data formats which seem possible to integrate—in other words, just because it is *possible* to merge two knowledge graphs that does not mean they should be!

The FAIR community has a long history of developing metadata standards, ontologies and provenance models. Research domains have also developed specific vocabularies and formats for repository submissions (often CSV-based), and likewise domain-specific models for making their registered data available as FAIR resources (often RDF-based, now more frequently JSON).

If we consider the lessons of evaluating FAIR Digital Objects and Linked Data in Chapter 3 on page 29, and the philosophy of RO-Crate from Chapter 4 on page 75, then it would seem important to facilitate proliferation of existing community standards, but also make their content more generally Findable and Accessible using a common overlay.

An approach that we have found useful with RO-Crate is therefore to propagate the existing provenance and metadata serializations, but also annotate their format and profiles in the RO-Crate metadata as not all formats are self-describing or well-known. General metadata (e.g. authors, license, subject) can then be extracted and replicated in the RO-Crate, increasing its coverage of the domain and making the metadata available to a wider set of technologies.

One approach for this covered by section Section 5.4.4.2 on page 175 describes how the PROV-based Common Provenance Model [Wittner 2023a] is used together with RO-Crate, both as a container of identified PROV bundles and by replicating the overall computation structure in the WRROC profiles. The full details are left in the PROV serialization that is carried along within the crate, and can be combined with distributed provenance of real-life processes such as transferring a biosample between a hospital and a lab.

Likewise interoperability with existing models is important, and Section 5.4.5.2 on page 179 showed how WRROC provenance can be mapped back to PROV. Note that some of the workflow details could be lost or muddled in a generic mapping if they don't have a corresponding pattern in PROV—for instance the expression of the workflow engine execution does not explicitly type it as such, that would require a particular PROV extension such as OPMW-PROV [Garijo 2011].

6.1.4.7 A strong community trumps semantically correctness

The development of Workflow Run Crate was done as a community activity (Section 5.4.6 on page 182), following the same pattern as RO-Crate itself (Section 4.1.2.6 on page 88) and many activities within the ELIXIR Europe life science network [Harrow 2022].

When collectively building semantic models, particularly using ontology design patterns [Hitzler 2016], it can often take much longer to figure out what is the *meaning* of a term (e.g. its semantics) rather than how it should be formalised in an ontology language. In research domains this often comes down to considering redefining core concepts of the field itself, which in life sciences for instance easily turn into philosophical dialectic arguments [Falk 2010].

The recently started Workflows Community Initiative has a working group for FAIR computational workflows[60], but before it is able to formalise the FAIR principles for workflows (building on [Goble 2020]), the group had to discuss to length what is or is not a *workflow*, and what makes it different from other Research Software.

While such fundamentals are important to get right, they should not become blockers for community progress. In the pragmatic take by the WRROC community for instance, a reverse argument was made that it is not so important if a workflow engine exists or not, but rather that we wanted to capture ”workflowy” provenance. The principle of “I know it when I see it” does not just apply to censorship of obscene material [Gewirtz 1996], but also to semantic design. In designing WRROC and RO-Crate it was useful to be constrained by the [schema.org] vocabulary, for instance the type http://schema.org/HowTo—with current examples showing how to change tires on a car—was also found adequate for describing a computational workflow and its steps in the Provenance Crate Profile (Section 5.4.2.3 on page 161).

This forced generalization may also have helped to make the model general enough for the different forms of workflow systems who implemented it, as each would have to “squint” slightly to map the WRROC concepts to their much more specific engine concepts. This also leads to fruitful community discussions and allowing reinterpretions of existing assumptions, placing the “just enough Linked Data” idea (Section 6.1.2.1 on page 188) into practice.

[60] https://workflows.community/groups/fair/

6.2 Conclusions

In this thesis I examined how to implement FAIR Research Objects and Computational Workflows by building on and simplifying approaches from Linked Data.

From research question **RQ1** (Section 1.2.1 on page 9) I asked if the FAIR Digital Object (FDO) concept was realisable using existing Web technology, which was explored by Chapters 2 and 3 and discussed in Section 6.1. The conclusion is that Web approaches can practically achieve FDO goals, by combining existing standards. However FAIR practitioners cannot simply equate Semantic Web with FDO, but need to also ensure sufficient constraints to guarantee navigational machine-actionability, balanced against developer usability and extensibility.

For research question **RQ2** (Section 1.2.2 on page 9) I endevour further on the challenge mentioned above: Chapter 4 introduced *RO-Crate* as such a pragmatic and normative approach that has been implemented by multiple open source developers for a wide range of applications. As discussed in Section 6.1, to build a reliable and extensible FDO ecosystem, the lightweight recommendations of RO-Crate needs to be combined with community-developed profiles which provide validation and tailored user interfaces. RO-Crate has been implemented by a wide range of Research Software, showing the learning curve for Linked Data can be reduced by use of common example-driven profiles and open collaborations.

In answer to the final research question **RQ3** (Section 1.2.3 on page 10), Chapter 5 covered different aspects, by proposing research software wrapped as canonical workflow building blocks, incrementally building FDOs from a workflow system, and recording workflow execution provenance. All of these have in common that they are implemented using RO-Crate and compatible with the other approaches from Chapter 4, e.g. for visualisation and editing. For workflow provenance, Section 5.4 on page 152 introduced the Workflow Run RO-Crate (WRROC) profiles. These were implemented by six different Workflow Management Systems, with maturing tooling and practical use cases that showcase how different developers (many not familiar with Semantic Web technologies) can adopt an interoperable approach to FDOs using pragmatic guidance to Linked Data.

From the considerations of this thesis I therefore conclude that FAIR Digital Objects with computational methods can be achieved using Web-based technologies, and can be implemented by Research Software Engineers across research domains without detailed training or experience with Linked Data technologies. This is promising for realising the full potential of digitally supported research with machine-actionable reproducible scholarly outputs.

A
Acknowledgements

A.1 Personal acknowledgements

This thesis would not have been possible without the tremendous support I have been so incredibly lucky to get from my closest and dearest throughout my PhD journey.

First of all I must thank my wonderful and amazing wife Gaby. You are my strongest pillar, making me motivated and focused, supporting me even when I have been difficult or absent-minded. I could not have done this without your love, and I am forever grateful. ¡Te amo con todo mi corazón!

Much of my daily joy comes from my children, and I am glad for all the time we have together—including during COVID-19 lockdown when work, study and life mixed all up! (apologies to readers if you find any Odd Squad references). You will probably insist I also thank Waldo—OK, he has also been there, although mostly sleeping in his little corner!

I want to thank my family in Norway—specially my parents who have been encouraging me throughout, and my siblings and cousins who always make it a pleasure to "come home" and give me regular updates when I'm away. Thank you to all my friends supporting me, in particular Laura, Juan, Ian, Rob, Sverre, Erlend, Magnus, André, Eline, Siv, Ove, Arne, Martin, Tonny, Øivind, Cecilie, Dag Rune, Ellen, Eli.

I am forever in debt to Paul Groth and I want to thank you for agreeing to supervise me as your PhD student, which must have been challenging at the best of times. I have enjoyed working with you since we first started talking about provenance and workflows more than a decade ago! Thank you for all your patience, persistence and for your pragmatic directions that kept me on track. Thanks also to the INDE lab[1] at University of Amsterdam, particularly for the daily colleaguality we got during lockdown—I wish I could have spent more time with you in person!

Thank you to Carole Goble, you have always been supporting me in so many ways since I started working for you in 2006, and you convinced me to start this PhD in the first place! No wonder you kept asking if I was finished yet! I appreciate our friendship and how you have mentored me to grow from a geeky developer to an acedemic. I hope we continue our exciting collaborations as "proper" colleagues!

I am so happy to have worked at the eScience Lab[2] at The University of Manchester for all these years and during this PhD. I particularly want to thank Shoaib Sufi for your wisdom and our deep conversations, not to mention your reliability whenever I have been stunned by bureaucracy or other struggles—I've probably been the oddest person for you to line manage! Thank you to all my amazing colleagues from the eScience Lab—specially Stuart, Finn, Aleks, Nick, Munazah, Alan, Doug, Oliver, Eli, Phil, Aleksander—you are a pleasure to work with, not just for your skills and expertise, but also your friendship and support.

I want to thank Meznah, you have been in the odd position of being co-supervised by me for your own PhD, and at the same time pushing me to finish mine! You bring such a joy with

[1] https://indelab.org/people/

[2] https://esciencelab.org.uk/people/

everything you do! Thanks to my fellow PhD and EngD students in Manchester: Fuqi, Ebitsam, Yo—you inspire me! Thanks to Rudolf Wittner, your productive provenance visits to Manchester have hopefully helped your PhD as much as mine!

I also want to thank Andrew Stewart, Uli Sattler and the rest of Department of Computer Science, who have been so accommodating, and you are now welcoming me again as I step into the academic world fully. It's not like you didn't prepare me! I also want to thank the UCU union members in Manchester (UMUCU)[3]—particularly Bijan and Ben for their personal support. I may have lost track of how many times we've been on the picket line during this PhD (no, I didn't manage to do much writing on strike days), but you have also showed me not to give up! Thank you to the engaging University of Manchester Students' Union, for standing up for what is right!

A.2 Community acknowledgements

I am grateful for all the wonderful discussions, technical contributrions and long-lasting friendships formed in the RO-Crate community (listed on page 213) since its early inception [Ó Carragáin 2019a] at Workshop on Research Objects (RO2018)[4]. Without you there would not have been any RO-Crate!

I would particularly like to thank Peter Sefton, without whose co-chairing, enthusiasm and pragmatism I would easily had got lost in the weeds. I am forever in debt to Eoghan Ó Carragáin who managed to get me and Peter to agree even on the trickiest issues, and led the community through its formative years.

I want to thank Simone Leo, whose persistent work and leadership on Workflow Run Crate[5] and the RO-Crate Python library [De Geest 2023a] has shown that RO-Crate works in practice even for detailed provenance.

Thank you to Bruno Kinoshita who kindly (and on his own initiative) helped proof this whole thesis, you have contributed to our common goals in so many ways, going back to when we first worked on Semantic Web, Apache Jena, and Common Workflow Language and later RO-Crate.

I am very grateful to Daniel Garijo, with whom I have enjoyed many insightful discussions and idea developments for more than a decade (since the early days of Research Objects!), from the cellar of Dagstuhl to beaches of San Diego! I appreciate your enthusiasm and many invaluable and constructive contributions to manuscripts, specifications and code.

I want to thank Leyla Jael Castro, for your continuous positivity and drive, and for pushing us to run RO-Crate training workshops together with Núria Queralt Rosinach, Claus Weiland and Jonas Grieb. I always appreciate our discussions and writing together, and I am so glad that you are much better than me to remember submission deadlines!

[3] https://manchester.web.ucu.org.uk/

[4] https://www.researchobject.org/ro2018/

[5] https://w3id.org/ro/wfrun/

RO-Crate is standing on the shoulders of giants, and I would like to thank the whole Research Object community [Goble 2018] for persisting on the early RO ideas [Newman 2009, Bechhofer 2013]. I have particular fond memories of whiteboard sessions with Sean Bechhofer, Khalid Belhajjame, Paolo Missier, David De Roure, and Kevin Page in the productive Wf4Ever project[6] that laid the foundation for what is now RO-Crate.

This PhD is in many ways born out of ELIXIR Europe[7], not just through the many fruitful co-authorships (Section B.1 on page 228), but also as the "ELIXIR way" of working is emblematic for the approach taken by RO-Crate community: open collaborations where institutional and project boundaries are blurred out in favour of jointly building pragmatic solutions.

I would like to thank the lovely people of ELIXIR Belgium (incl. Frederik, Ignacio, Paul, Rafael, Bert), ELIXIR Spain (incl. José Mª, Laura, Salva, Adam, Pau), ELIXIR HUB (incl. Justin, Jonathan, Marieke, Gavin, Niklas, Katharina), ELIXIR DE (incl. Björn, Anika, Sebastian, Nils, Michael), ELIXIR CH (Alex, Patricia), ELIXIR UK (incl. Susanna, Nicola, Xenia, Tim, Neil), ELIXIR NL (Núria, Egon, Chris, Marco, Rob, Helena) and many others I may have missed listing. I want to particularly thank everyone who organized and participated in the ELIXIR Biohackathons[8] which always inspires me and has helped build our communities.

This thesis frequently mentions the Common Workflow Language[9], and many of the ideas spur out of methods that were first tried by the CWL community[10] (see for example [Möller 2017, Robinson 2017, Khan 2019, Crusoe 2022]), which I am grateful for. I want to thank Michael Crusoe, not just for your tireless work on bringing CWL together and willingness to put FAIR workflow theories into practice, but also for our enduring friendship and your support throughout most of the rabbit holes I have ventured into. I want to thank Peter Amstutz, Hervé Ménager and the rest of the CWL leadership team and the many volunteers.

I also need to thank members of the BioCompute Object[11] community, particularly Hadley, Jonathon, Raja, Dennis, Vahan, Janisha, Amanda, and Nicola. I thank the Workflows Community Initiative[12] in particular Sean R. Wilkinson, Rafael Ferreira da Silva, and Kyle Chard.

From the FAIR Digital Object Forum[13] I need to place particular thanks to Peter Wittenburg, Luiz Olavo Bonino da Silva Santos, Maggie Hellström, Christophe Blanchi, Ulrich Schwardmann and Rainer Stotzka. I also want to thank all the members of the Research Data Alliance[14]'s FAIR Digital Object Fabric interest group.

Mark Wilkinson, we first worked together on BioMoby services and Taverna workflows, which seems a long way from where we are today with FAIR! Looking back from this thesis, perhaps

[6] https://s11.no/2020/archive/wf4ever/
[7] https://elixir-europe.org/
[8] https://biohackathon-europe.org/
[9] About 97 times!
[10] https://www.commonwl.org/contributors/
[11] https://biocomputeobject.org/
[12] https://workflows.community/
[13] https://fairdo.org/
[14] https://www.rd-alliance.org/

we could consider those to be early FAIR digital objects as well? I want to thank you for your enthusiasm, and for bringing me in to the Apples2Apples hackathons—thank you also to Robert, Richard, Wilko, David, Alban, and Allyson.

Many thanks to Sarven Capadisli, you have inspired me in so many ways, from our long evening rants about Linked Data, scholarly communications and COVID-19; through your and Amy Guy[15]'s persistence on using the Web's full capabilities for Linked Research[16] (which encouraged me to make the Web version of this thesis[17]), to welcoming me to stay with your family and feeding me chillis that were too hot even for this pretend-Mexican!

Thank you to Egon Willighagen, I always enjoy our conversations, be it in the wind at Zandvoort, at the warmth of our Biohackathon cabin, or in long-winded digressions on social media. I am grateful for your help, motivation and Dutch expertise (B.9 on page 322).

I am indebted to Herbert van de Sompel—our conversations on the topic of aggregations, annotations, mementos and the Web started more than a decade ago; your clear ideas and powerful visions are at the heart of what this thesis tries to achieve. I am always energised by your talks, and truly appreciate working with you to promote FAIR Signposting and hope we will continue such exciting work together.

A.2.1 RO-Crate Community

As of 2024-04-29, the *RO-Crate* Community members[18] are:

Peter Sefton	https://orcid.org/0000-0002-3545-944X	(co-chair)
Stian Soiland-Reyes	https://orcid.org/0000-0001-9842-9718	(co-chair)
Eoghan Ó Carragáin	https://orcid.org/0000-0001-8131-2150	(emeritus chair)
Oscar Corcho	https://orcid.org/0000-0002-9260-0753	
Daniel Garijo	https://orcid.org/0000-0003-0454-7145	
Raul Palma	https://orcid.org/0000-0003-4289-4922	
Frederik Coppens	https://orcid.org/0000-0001-6565-5145	
Carole Goble	https://orcid.org/0000-0003-1219-2137	
José María Fernández	https://orcid.org/0000-0002-4806-5140	
Kyle Chard	https://orcid.org/0000-0002-7370-4805	
Jose Manuel Gomez-Perez	https://orcid.org/0000-0002-5491-6431	

[15] https://dr.amy.gy/
[16] https://csarven.ca/linked-research-decentralised-web
[17] https://s11.no/2023/phd/
[18] https://www.researchobject.org/ro-crate/community.html

Michael R Crusoe	https://orcid.org/0000-0002-2961-9670
Ignacio Eguinoa	https://orcid.org/0000-0002-6190-122X
Nick Juty	https://orcid.org/0000-0002-2036-8350
Kristi Holmes	https://orcid.org/0000-0001-8420-5254
Jason A. Clark	https://orcid.org/0000-0002-3588-6257
Salvador Capella-Gutierrez	https://orcid.org/0000-0002-0309-604X
Alasdair J. G. Gray	https://orcid.org/0000-0002-5711-4872
Stuart Owen	https://orcid.org/0000-0003-2130-0865
Alan R Williams	https://orcid.org/0000-0003-3156-2105
Giacomo Tartari	https://orcid.org/0000-0003-1130-2154
Finn Bacall	https://orcid.org/0000-0002-0048-3300
Thomas Thelen	https://orcid.org/0000-0002-1756-2128
Hervé Ménager	https://orcid.org/0000-0002-7552-1009
Laura Rodríguez-Navas	https://orcid.org/0000-0003-4929-1219
Paul Walk	https://orcid.org/0000-0003-1541-5631
brandon whitehead	https://orcid.org/0000-0002-0337-8610
Mark Wilkinson	https://orcid.org/0000-0001-6960-357X
Paul Groth	https://orcid.org/0000-0003-0183-6910
Erich Bremer	https://orcid.org/0000-0003-0223-1059
LJ Garcia Castro	https://orcid.org/0000-0003-3986-0510
Karl Sebby	https://orcid.org/0000-0001-6022-9825
Alexander Kanitz	https://orcid.org/0000-0002-3468-0652
Ana Trisovic	https://orcid.org/0000-0003-1991-0533
Gavin Kennedy	https://orcid.org/0000-0003-3910-0474
Mark Graves	https://orcid.org/0000-0003-3486-8193
Jasper Koehorst	https://orcid.org/0000-0001-8172-8981
Simone Leo	https://orcid.org/0000-0001-8271-5429
Marc Portier	https://orcid.org/0000-0002-9648-6484
Paul Brack	https://orcid.org/0000-0002-5432-2748

Milan Ojsteršek	https://orcid.org/0000-0003-1743-8300
Bert Droesbeke	https://orcid.org/0000-0003-0522-5674
Chenxu Niu	https://orcid.org/0000-0002-2142-1731
Kosuke Tanabe	https://orcid.org/0000-0002-9986-7223
Tomasz Miksa	https://orcid.org/0000-0002-4929-7875
Marco La Rosa	https://orcid.org/0000-0001-5383-6993
Cedric Decruw	https://orcid.org/0000-0001-6387-5988
Andreas Czerniak	https://orcid.org/0000-0003-3883-4169
Jeremy Jay	https://orcid.org/0000-0002-5761-7533
Sergio Serra	https://orcid.org/0000-0002-0792-8157
Ronald Siebes	https://orcid.org/0000-0001-8772-7904
Shaun de Witt	https://orcid.org/0000-0003-4196-3658
Shady El Damaty	https://orcid.org/0000-0002-2318-4477
Douglas Lowe	https://orcid.org/0000-0002-1248-3594
Xuanqi Li	https://orcid.org/0000-0003-1498-6205
Sveinung Gundersen	https://orcid.org/0000-0001-9888-7954
Muhammad Radifar	https://orcid.org/0000-0001-9156-9478
Rudolf Wittner	https://orcid.org/0000-0002-0003-2024
Oliver Woolland	https://orcid.org/0000-0002-4565-9760
Paul De Geest	https://orcid.org/0000-0002-8940-4946
Douglas Fils	https://orcid.org/0000-0002-2257-9127
Florian Wetzels	https://orcid.org/0000-0002-5526-7138
Raül Sirvent	https://orcid.org/0000-0003-0606-2512
Abigail Miller	https://orcid.org/0000-0001-9228-2882
Jake Emerson	https://orcid.org/0000-0003-0617-9219
Davide Fucci	https://orcid.org/0000-0002-0679-4361
Bruno P. Kinoshita	https://orcid.org/0000-0001-8250-4074
Maciek Bąk	https://orcid.org/0000-0003-1361-7301
Jens Hollunder	https://orcid.org/0000-0003-3234-6762

Martin Weise	http://orcid.org/0000-0003-4216-302X
Vartika Bisht	https://orcid.org/0000-0002-1880-0597
Toshiyuki Nishiyama Hiraki	https://orcid.org/0000-0001-6712-6335
Bram Ulrichts	https://orcid.org/0000-0002-5934-8998
Michael Falk	https://orcid.org/0000-0001-9261-8390
Eli Chadwick	https://orcid.org/0000-0002-0035-6475
Daniel Bauer	https://orcid.org/0000-0001-9447-460X
James Love	https://orcid.org/0000-0001-7760-1240

A.2.2 In remembrance

I am saddened by the loss of Olav S. Bratli[19], Stu Allan[20], James Taylor[21], Sarah Jones[22], Henry Story[23], Chris Connolly[24], Lloyd Cawthorne[25], and other such brilliant and lovely people.

In memoriam: Damon Harvey, Luke O'Connor, Cian Chantrill, Brianna Ghey, Ben Trueman, Ryan Watson, Finn Kitson, Rory Wood, William King, Harrison De George, Charlotte Burlace-Colquhoun, Finn Kitson, Charley Gadd, Sumanta Bansi, Rhys Hill, Laura Nuttall and other students whose lives were cut too short.

My thoughts are with the victims of war, conflict, crime and abuse in Ukraine, Palestine, Israel, Yemen, Sudan, Mexico and across the world.

[19] https://www.youtube.com/watch?v=y8EhezA5_LY

[20] https://stuallan.co.uk/

[21] https://galaxyproject.org/jxtx/

[22] https://www.rd-alliance.org/remembering-sarah-jones

[23] https://lists.w3.org/Archives/Public/semantic-web/2023Sep/0015.html

[24] https://staffnet.cs.manchester.ac.uk/newsletters/CS_Newsletter_2014-03-31.htm

[25] https://mancunion.com/2022/12/13/make-sure-it-never-happens-again-students-gather-in-memory-of-dr-lloyd-cawthorne/

A.3 My funding

My work presented in this thesis has been undertaken during several research projects at The University of Manchester, which funding is acknowledged below. I am grateful for all the discussions and collaborations in these projects.

European Commission programme Horizon H2020

H2020-INFRAEDI-02-2018	823830[26]	BioExcel-2[27]
H2020-INFRAEOSC-2018-2	824087[28]	EOSC-Life[29]
H2020-INFRAIA-2017-1-two-stage	730976[30]	IBISBA 1.0[31]
H2020-INFRAIA-2018-1	823827[32]	SyntheSys+[33]

European Commission programme Horizon Europe			**UKRI**[34]
HORIZON-INFRA-2021-EMERGENCY-01	101046203[35]	BY-COVID[36]	
HORIZON-INFRA-2021-EOSC-01	101057388[37]	EuroScienceGateway[38]	10038963[39]
HORIZON-INFRA-2021-TECH-01	101057437[40]	BioDT[41]	10038930[42]
HORIZON-INFRA-2021-EOSC-01-05	101057344[43]	FAIR-IMPACT[44]	10038992[45]

UK Research and Innovation (UKRI)

MRC / DARE-UK[46][47]	TRE-FX[48]	MC_PC_23007[49]

[26] https://doi.org/10.3030/823830
[27] https://bioexcel.eu/
[28] https://doi.org/10.3030/824087
[29] https://www.eosc-life.eu/
[30] https://doi.org/10.3030/730976
[31] https://ibisba.eu/
[32] https://doi.org/10.3030/823827
[33] https://www.synthesys.info/
[34] UK Research and Innovation under the UK government's *Horizon Europe funding guarantee*
[35] https://doi.org/10.3030/101046203
[36] https://by-covid.eu/
[37] https://doi.org/10.3030/101057388
[38] http://eurosciencegateway.eu/
[39] https://gtr.ukri.org/projects?ref=10038963
[40] https://doi.org/10.3030/101057437
[41] https://biodt.eu/
[42] https://gtr.ukri.org/projects?ref=10038930
[43] https://doi.org/10.3030/101057344
[44] https://fair-impact.eu/
[45] https://gtr.ukri.org/projects?ref=10038992

A.4 Funding and acknowledgements for co-authored chapters

A.4.1 Acknowledgements for *Evaluating FAIR Digital Object and Linked Data as distributed object systems*

Sections 2 on page 16 and 3.1 on page 31 are adapted from a journal article accepted for publication in PeerJ Computer Science.

Published As

Stian Soiland-Reyes, Carole Goble, Paul Groth (2024):
Evaluating FAIR Digital Object and Linked Data as distributed object systems.
PeerJ Computer Science **10**:e1781 (accepted)
https://doi.org/10.7717/peerj-cs.1781

An RO-Crate for this article[50] is archived in Zenodo [Soiland-Reyes 2023a].

Acknowledgements

We would like to acknowledge the FAIR Digital Object Forum[51] community and working groups, where SSR and CG are members.

Views and opinions expressed in this work are those of the authors only and do not necessarily reflect those of the funded projects, FAIR Digital Object Forum, European Union nor the European Commission.

License and modifications

- **License**: Creative Commons Attribution License (CC BY 4.0).
- **Modifications**: Minor LaTeX changes; references in s11 house style; citations merged and renumbered; funding and references moved to separate chapters. Background moved to separate chapter, added footnote to define *machine actionable*, added section 2.1.1, figures 2.1, 2.2 and listing 2.1. Minor typos and grammatical errors fixed. Added glossary/acronym links.

Funding

This work was funded by the European Union programmes *Horizon 2020* under grant agreements H2020-INFRAEDI-02-2018 823830 (BioExcel-2), H2020-INFRAEOSC-2018-2 824087 (EOSC-Life)

[46] https://dareuk.org.uk/driver-project-tre-fx/
[47] Call: *Inform design of cross-council trusted research environments*
[48] https://trefx.uk/
[49] https://gtr.ukri.org/projects?ref=MC_PC_23007
[50] https://w3id.org/ro/doi/10.5281/zenodo.8075229
[51] https://fairdo.org/

and *Horizon Europe* under grant agreements HORIZON-INFRA-2021-EMERGENCY-01 101046203 (BY-COVID), HORIZON-INFRA-2021-EOSC-01 101057388 (EuroScienceGateway), HORIZON-INFRA-2021-EOSC-01-05 101057344 (FAIR-IMPACT), HORIZON-INFRA-2021-TECH-01 101057437 (BioDT), HORIZON-CL4-2021-HUMAN-01-01 101070305 (ENEXA); and by UK Research and Innovation (UKRI) under the UK government's *Horizon Europe funding guarantee* grants 10038963 (EuroScienceGateway), 10038992(FAIR-IMPACT), 10038930 (BioDT).

The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

A.4.2 Acknowledgements for *Updating Linked Data practices for FAIR Digital Object principles*

Section 3.2 is adapted from a published peer-reviewed conference abstract, presented as talk by Stian Soiland-Reyes at First International Conference on FAIR Digital Objects[52] (FDO2022) on 2022-08-26/–28 in Leiden, The Netherlands.

- Slides: https://doi.org/10.5281/zenodo.7256428

Published As

Stian Soiland-Reyes, Leyla Jael Castro, Daniel Garijo, Marc Portier, Carole Goble, Paul Groth (2022):
Updating Linked Data practices for FAIR Digital Object principles.
1st International Conference on FAIR Digital Objects (FDO 2022) (abstract).
Research Ideas and Outcomes **8**:e94501
https://doi.org/10.3897/rio.8.e94501

License and modifications

- **License**: Creative Commons Attribution License (CC BY 4.0).
- **Modifications**: Formatting as Markdown and LaTeX; figure caption formatting; reference in s11 house style[53]; citations merged and renumbered; new introduction; acknowledgements and references moved to separate chapters, fixed minor typos and grammatical errors. Added glossary/acronym links.

Acknowledgements

We would like to acknowledge the RO-Crate community[54] and the WorkflowHub Club[55]. Thanks to Rudolf Wittner for valuable comments.

[52] https://www.fdo2022.org/
[53] https://s11.no/2021/house-rules/citation-style/
[54] https://www.researchobject.org/ro-crate/community.html
[55] https://about.workflowhub.eu/project/acknowledgements/

Funding

European Commission Horizon 2020 (EOSC-Life 824087[56]), Horizon Europe (BY-COVID 101046203[57], FAIR-IMPACT 101057344[58]).

Leyla Jael Castro is supported by a German Research Foundation DFG grant for NFDI4DataScience.

Daniel Garijo is supported by the Madrid Government (Comunidad de Madrid-Spain) under the Multiannual Agreement with Universidad Politécnica de Madrid in the line Support for R&D projects for Beatriz Galindo researchers, in the context of the V PRICIT (Regional Programme of Research and Technological Innovation)

A.4.3 Acknowledgements for *Packaging research artefacts with RO-Crate*

Sections 4.1 on page 77 and 4.3 on page 113 are adapted from an article published in the journal *Data Science*.

Published As

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 (2022):
Packaging research artefacts with RO-Crate.
Data Science **5**(2)
https://doi.org/10.3233/DS-210053

An RO-Crate for this article[59] is archived in Zenodo [Soiland-Reyes 2022g].

License and modifications

- **License**: Creative Commons Attribution License (CC BY 4.0).
- **Modifications**: Formatting as Markdown and LaTeX; figure caption formatting; reference in s11 house style; added identifiers, authors and years clarified where missing in citations; inline citation hyperlinks to open access version where available; citations merged and renumbered; acknowledgements and references moved to separate chapters; fixed minor typos and grammatical errors. Added glossary/acronym links.

[56] https://doi.org/10.3030/824087
[57] https://doi.org/10.3030/101046203
[58] https://doi.org/10.3030/101057344
[59] https://w3id.org/ro/doi/10.5281/zenodo.5146227

Funding

This work has received funding from the European Commission's Horizon 2020 research and innovation programme for projects BioExcel-2[60] (H2020-INFRAEDI-2018-1 823830), IBISBA 1.0[61] (H2020-INFRAIA-2017-1-two-stage 730976), PREP-IBISBA[62] (H2020-INFRADEV-2019-2 871118), EOSC-Life[63] (H2020-INFRAEOSC-2018-2 824087), SyntheSys+[64] (H2020-INFRAIA-2018-1 823827). From the Horizon Europe Framework Programme this work has received funding for BY-COVID[65] (HORIZON-INFRA-2021-EMERGENCY-01 101046203).

Björn Grüning is supported by DataPLANT (NFDI 7/1 – 42077441)[66], part of the German National Research Data Infrastructure (NFDI), funded by the Deutsche Forschungsgemeinschaft (DFG).

Ana Trisovic is funded by the Alfred P. Sloan Foundation. (Grant number P-2020-13988)[67] Harvard Data Commons is supported by an award from Harvard University Information Technology (HUIT).

A.4.4 Acknowledgements for *Creating lightweight FAIR Digital Objects with RO-Crate*

Section 4.2 on page 109 is adapted from a published peer-reviewed conference abstract, presented as poster by Stian Soiland-Reyes at First International Conference on FAIR Digital Objects (FDO2022)[68] on 2022-08-26/–28 in Leiden, The Netherlands.

- Poster: https://doi.org/10.5281/zenodo.7245315

Published As

Stian Soiland-Reyes, Peter Sefton, Leyla Jael Castro, Frederik Coppens, Daniel Garijo, Simone Leo, Marc Portier, Paul Groth (2022):
Creating lightweight FAIR Digital Objects with RO-Crate.
1st International Conference on FAIR Digital Objects (FDO2022) (poster)
Research Ideas and Outcomes **8**:e93937
https://doi.org/10.3897/rio.8.e93937

License and modifications

- **License**: Creative Commons Attribution License (CC BY 4.0).

[60] https://doi.org/10.3030//823830
[61] https://doi.org/10.3030/730976
[62] https://doi.org/10.3030/871118
[63] https://doi.org/10.3030/824087
[64] https://doi.org/10.3030/823827
[65] https://doi.org/10.3030/101046203
[66] https://gepris.dfg.de/gepris/projekt/442077441
[67] https://sloan.org/grant-detail/9555
[68] https://www.fdo2022.org/

- **Modifications**: Formatting as Markdown and LaTeX; figure caption formatting; references in s11 house style; citations merged and renumbered; acknowledgements and references moved to separate chapters. Figure 4.5 font re-rendered. Added Figure 4.6. Fixed minor typos and grammatical errors. Added glossary/acronym links.

Acknowledgements

We would like to acknowledge the RO-Crate community[69] and the WorkflowHub Club[70].

Funding

European Commission Horizon 2020 (BioExcel-2 823830[71], EOSC-Life 824087[72]), Horizon Europe (BY-COVID 101046203[73], FAIR-IMPACT 101057344[74]).

Daniel Garijo is supported by the Madrid Government (Comunidad de Madrid-Spain) under the Multiannual Agreement with Universidad Politécnica de Madrid in the line Support for R&D projects for Beatriz Galindo researchers, in the context of the V PRICIT (Regional Programme of Research and Technological Innovation).

Leyla Jael Castro is supported by a German Research Foundation DFG grant for NFDI4DataScience.

Frederik Coppens is supported by Research Foundation - Flanders (FWO) for ELIXIR Belgium (I002819N).

A.4.5 Acknowledgements for *Making Canonical Workflow Building Blocks*

Section 5.1 on page 121 is adapted from journal article published in *Data Intelligence*.

Published As

Stian Soiland-Reyes, Genís Bayarri, Pau Andrio, Robin Long, Douglas Lowe, Ania Niewielska, Adam Hospital, Paul Groth (2022):
Making Canonical Workflow Building Blocks interoperable across workflow languages.
Data Intelligence **4**(2)
https://doi.org/10.1162/dint_a_00135

License and modifications

- **License**: Creative Commons Attribution License (CC BY 4.0).

[69] https://www.researchobject.org/ro-crate/community.html
[70] https://about.workflowhub.eu/project/acknowledgements/
[71] https://doi.org/10.3030/823830
[72] https://doi.org/10.3030/824087
[73] https://doi.org/10.3030/101046203
[74] https://doi.org/10.3030/101057344

- **Modifications**: Formatting as Markdown and LaTeX; figure caption formatting; references in s11 house style; citations merged and renumbered; acknowledgements and references moved to separate chapters. Fixed minor typos and grammatical errors. Added glossary/acronym links.

Acknowledgements

This work has been done as part of the BioExcel CoE (https://www.bioexcel.eu/), a project funded by the European Union contracts H2020-INFRAEDI-02-2018 823830[75], H2020-EINFRA-2015-1 675728[76]. Additional work is funded through EOSC-Life (https://www.eosc-life.eu/) contract H2020-INFRAEOSC-2018-2 824087[77], and ELIXIR-CONVERGE (https://elixir-europe.org/) contract H2020-INFRADEV-2019-2 871075[78].

The authors would also like to acknowledge contributions from: Felix Amaladoss, Cibin Sadasivan Baby, Finn Bacall, Rosa M. Badia, Sarah Butcher, Gerard Capes, Michael R. Crusoe, Alberto Eusebi, Carole Goble, Josep Lluís Gelpí, Modesto Orozco, Geoff Williams, Felix Amaladoss

A.4.6 Acknowledgements for *The Specimen Data Refinery*

Section 5.2 on page 133 is adapted from a journal article published in *Data Intelligence*.

Published As

Alex Hardisty, Paul Brack, Carole Goble, Laurence Livermore, Ben Scott, Quentin Groom, Stuart Owen, Stian Soiland-Reyes (2022):
The Specimen Data Refinery: A canonical workflow framework and FAIR Digital Object approach to speeding up digital mobilisation of natural history collections.
Data Intelligence **4**(2)
https://doi.org/10.1162/dint_a_00134

License and modifications

- **License**: Creative Commons Attribution License (CC BY 4.0).
- **Modifications**: Formatting as Markdown and LaTeX; figures replaced with higher resolutions from source; figure caption formatting; references in s11 house style; added DOIs and URLs; cited preprints replaced with later publications citations merged and renumbered; funding and references moved to separate chapters; fixed minor typos and grammatical errors; removed citation Speicher 2015 (*Linked Data*, not *Linked Data Platform*); added glossary/acrony links.

[75] https://doi.org/10.3030/823830
[76] https://doi.org/10.3030/675728
[77] https://doi.org/10.3030/824087
[78] https://doi.org/10.3030/871075

Acknowledgements

This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement numbers 823827 (SYNTHESYS Plus), 871043 (DiSSCo Prepare), 823830 (BioExcel-2), 824087 (EOSC-Life).

A.4.7 Acknowledgements for *Incrementally building FAIR Digital Objects*

Section 5.3 on page 148 is adapted from an abstract presented as poster by Stian Soiland-Reyes at First International Conference on FAIR Digital Objects[79] (FDO2022) on 2022-08-26/–28 in Leiden, The Netherlands.

Published As

Oliver Woolland, Paul Brack, Stian Soiland-Reyes, Ben Scott, Laurence Livermore (2022):
Incrementally building FAIR Digital Objects with Specimen Data Refinery workflows.
1st International Conference on FAIR Digital Objects (FDO 2022) (poster)
Research Ideas and Outcomes **8**:e94349
https://doi.org/10.3897/rio.8.e94349

License and modifications

- **License**: Creative Commons Attribution License (CC BY 4.0).
- **Modifications**: Formatting as Markdown and LaTeX; references in s11 house style; added DOIs and URLs; funding and references moved to separate chapters; fixed minor typos and grammatical errors; added glossary/acronym links.

Acknowledgements

We acknowledge the SYNTHESYS+[80] and DiSSCO[81] project members who have been invaluable in early evaluation and feedback on the development of SDR.

Funding

This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement numbers 823827[82] (SYNTHESYS Plus), 871043[83] (DiSSCo Prepare), 823830[84] (BioExcel-2), 824087[85] (EOSC-Life).

[79] https://www.fdo2022.org/
[80] https://www.synthesys.info/
[81] https://www.dissco.eu/
[82] https://doi.org/10.3030/823827
[83] https://doi.org/10.3030/871043
[84] https://doi.org/10.3030/823830
[85] https://doi.org/10.3030/824087

A.4.8 Acknowledgement for *Recording provenance of workflow runs with RO-Crate*

Section 5.4 on page 152 is adapted from an arXiv preprint, revised manuscript published in PLOS One.

Published As

Simone Leo, Michael R. Crusoe, Laura Rodríguez-Navas, Raül Sirvent, Alexander Kanitz, Paul De Geest, Rudolf Wittner, Luca Pireddu, Daniel Garijo, José M. Fernández, Iacopo Colonnelli, Matej Gallo, Tazro Ohta, Hirotaka Suetake, Salvador Capella-Gutierrez, Renske de Wit, Bruno de Paula Kinoshita, Stian Soiland-Reyes (2024):
Recording provenance of workflow runs with RO-Crate.
PLOS One **19**(9):e0309210
arXiv 2312.07852 [cs.DL]
https://doi.org/10.48550/arXiv.2312.07852
https://doi.org/10.1371/journal.pone.0309210

An RO-Crate for this article[86] is archived in Zenodo [Leo 2023b].

License and modifications

- **License**: Creative Commons Attribution License (CC BY 4.0).
- **Modifications**: Formatting as Markdown and figure caption formatting; references in s11 house style; URLs as footnotes/hyperlinks; enumerations made explicit; listing captions; some paragraphs split for readability; details moved to footnote, acknowledgement and references moved to separate chapters; fixed minor typos and grammatical errors; added glossary/acronym links; acronym WfMS instead of WMS.

Acknowledgements

The authors would like to thank all participants to the Workflow Run RO-Crate working group[87] meetings for the positive detailed discussions and valuable feedback.

The authors acknowledge funding from: Sardinian Regional Government through the XData Project (S.L., L.P.); Spanish Government (contract PID2019-107255GB) (R.S.); MCIN/AEI/10.13039/501100011033 (CEX2021-001148-S) (R.S.); Generalitat de Catalunya (contract 2021-SGR-00412) (R.S.); European High-Performance Computing Joint Undertaking (JU) (No 955558[88]) (R.S.); EU Horizon research and innovation programme under Grant agreement No 101058129[89] (DT-GEO) (R.S.); ELIXIR Platform Task 2022-2023 funding for

[86] https://w3id.org/ro/doi/10.5281/zenodo.10368989
[87] https://www.researchobject.org/workflow-run-crate/#community
[88] https://doi.org/10.3030/955558
[89] https://doi.org/10.3030/101058129

Task “Container Orchestration” (A.K.); Research Foundation - Flanders (FWO) for ELIXIR Belgium (I000323N and I002819N) (P.D.G.); Multiannual Agreement with Universidad Politécnica de Madrid in the line Support for R&D projects for Beatriz Galindo researchers, in the context of the V PRICIT (Regional Programme of Research and Technological Innovation) (D.G.); Comunidad de Madrid through the call Research Grants for Young Investigators from Universidad Politécnica de Madrid (D.G.); ICSC - Centro Nazionale di Ricerca in High-Performance Computing, Big Data and Quantum Computing, funded by European Union - NextGenerationEU (I.C.); ACROSS project, HPC Big Data Artificial Intelligence Cross Stack Platform Towards Exascale, funded by the European High-Performance Computing Joint Undertaking (JU) under G.A. n. 955648[90] (I.C.); EUPEX project, European Pilot for Exascale, funded by the European High-Performance Computing Joint Undertaking (JU) under G.A. n. 101033975[91] (I.C.); Life Science Database Integration Project, NBDC of Japan Science and Technology Agency (T.O.); JSPS KAKENHI (Grant Number 20J22439[92]); European Commission Horizon 2020 H2020-SC1-2018-Single-Stage-RTD 825575[93] (European Joint Programme on Rare Diseases; SC1-BHC-04-2018 Rare Disease European Joint Programme Cofund) (L.R.N., J.M.F., S.C.G.), European High-Performance Computing Joint Undertaking (JU) (No 955558), EU NextGenerationEU/PRTR (project eFlows4HPC) H2020-JTI-EuroHPC-2019-1 955558[94] (eFlows4HPC) (R.S.), H2020-INFRAEDI-02-2018 823830[95] (BioExcel-2) (S.S.R.), H2020-INFRAEOSC-2018-2 824087[96] (EOSC-Life) (S.L., L.R.N., P.D.G., R.W., L.P., J.M.F., S.C.G., S.S.R.); Horizon Europe HORIZON-INFRA-2021-EMERGENCY-01 101046203[97] (BY-COVID) (S.L., L.R.N., P.D.G., R.W., L.P., J.M.F., S.C.G., S.S.R.), HORIZON-INFRA-2021-EOSC-01 101057388[98] (EuroScienceGateway) (P.D.G., J.M.F., S.C.G., S.S.R.), HORIZON-INFRA-2021-EOSC-01-05 101057344[99] (FAIR-IMPACT) (D.G., S.S.R.); UK Research and Innovation (UKRI) under the UK government’s Horizon Europe funding guarantee 10038963[100] (EuroScienceGateway), 10038992[101] (FAIR-IMPACT) (S.S.R.).

[90] https://doi.org/10.3030/955648

[91] https://doi.org/10.3030/101033975

[92] https://kaken.nii.ac.jp/en/grant/KAKENHI-PROJECT-20J22439/

[93] https://doi.org/10.3030/825575

[94] https://doi.org/10.3030/955558

[95] https://doi.org/10.3030/823830

[96] https://doi.org/10.3030/824087

[97] https://doi.org/10.3030/101046203

[98] https://doi.org/10.3030/101057388

[99] https://doi.org/10.3030/101057344

[100] https://gtr.ukri.org/projects?ref=10038963

[101] https://gtr.ukri.org/projects?ref=10038992

B

Contributions

Here I detail my contributions for each chapter of this thesis, and list all the other contributors and their affiliations.

B.1 Thesis contributions

Below are the author contributions to published articles that form part of this thesis. Contributions are classified primarily according to the Contributor Roles Taxonomy (CASRAI CrEDiT) [Brand 2015]. See also appendix A on page 210 for acknowledgements beyond authorship covered below.

For all chapters except Sections 5.2 and 5.4, I am the main author of the corresponding manuscripts and have contributed to all aspects of the research. See details below:

B.1.1 Contributions for *Evaluating FAIR Digital Object as a distributed object system*

Chapter 2 on page 15 and Section 3.1 on page 31 were co-authored by:

Stian Soiland-Reyes Conceived and designed the experiments, performed the experiments, analyzed the data, prepared figures and/or tables, authored or reviewed drafts of the article; contributed Conceptualization, Data Curation, Formal Analysis, Funding acquisition, Investigation, Methodology, Software, Writing – original draft, Writing – review and editing; and approved the final draft.

Carole Goble conceived and designed the experiments, authored or reviewed drafts of the article; contributed Funding acquisition, Supervision, Writing – review and editing; and approved the final draft.

Paul Groth conceived and designed the experiments, authored or reviewed drafts of the article; contributed Conceptualization, Methodology, Supervision, Writing – original draft, Writing – review and editing; and approved the final draft.

I am the main author of the corresponding manuscript and have contributed to all aspects of the research.

B.1.2 Contributions for *Updating Linked Data practices for FAIR Digital Object principles*

Section 3.2 on page 71 was co-authored by:

Stian Soiland-Reyes Conceptualization, Formal Analysis, Funding acquisition, Investigation, Software, Writing – original draft, Writing – review and editing

Leyla Jael Castro Writing – original draft

Daniel Garijo Conceptualization, Funding acquisition, Writing – review and editing

Marc Portier Investigation, Writing – original draft, Writing – review and editing

Carole Goble: Funding acquisition, Supervision

Paul Groth Supervision

This work was presented as talk by Stian Soiland-Reyes at First International Conference on FAIR Digital Objects, Leiden, The Netherlands.

- Slides: https://doi.org/10.5281/zenodo.7256428

B.1.3 Contributions for *Packaging research artefacts with RO-Crate*

Section 4.1 on page 77 was co-authored by:

Stian Soiland-Reyes Conceptualization, Data curation, Formal Analysis, Funding acquisition, Investigation, Methodology, Project administration, Software, Visualization, Writing – original draft, Writing – review & editing

Peter Sefton Conceptualization, Investigation, Methodology, Project administration, Resources, Software, Writing – review & editing

Mercè Crosas Writing – review & editing

Leyla Jael Castro Methodology, Writing – review & editing

Frederik Coppens Writing – review & editing

José M. Fernández Methodology, Software, Writing – review & editing

Daniel Garijo Methodology, Writing – review & editing

Björn Grüning Writing – review & editing

Marco La Rosa Software, Methodology, Writing – review & editing

Simone Leo Software, Methodology, Writing – review & editing

Eoghan Ó Carragáin Investigation, Methodology, Project administration, Writing – review & editing

Marc Portier Methodology, Writing – review & editing

Ana Trisovic Software, Writing – review & editing

RO-Crate Community[1] Investigation, Software, Validation, Writing – review & editing

Paul Groth Methodology, Supervision, Writing – original draft, Writing – review & editing

Carole Goble Conceptualization, Funding acquisition, Methodology, Project administration, Supervision, Visualization, Writing – review & editing

I am the main author and editor of the corresponding manuscript and have contributed to all aspects of the research. Section 4.1.4.1 on page 95 was primarily authored by Simone Leo. Section

4.1.4.4 on page 99 was primarily authored by Leyla Jael Castro. Section 4.1.4.5 on page 100 with Figure 4.4 was authored by Mercè Crosas and Ana Trisovic, and edited by me.

The co-authors would also like to acknowledge contributions from:

Finn Bacall Software, Methodology

Herbert Van de Sompel Writing – review & editing

Ignacio Eguinoa Software, Methodology

Nick Juty Writing – review & editing

Oscar Corcho Writing – review & editing

Stuart Owen Writing – review & editing

Laura Rodríguez-Navas Software, Visualization, Writing – review & editing

Alan R. Williams Writing – review & editing

B.1.4 Contributions for *Creating lightweight FAIR Digital Objects with RO-Crate*

Section 4.2 on page 109 was co-authored by:

Stian Soiland-Reyes Conceptualization, Funding acquisition, Project administration, Software, Writing – original draft, Writing – review & editing

Peter Sefton Funding acquisition, Project administration, Software

Leyla Jael Castro Writing – original draft, Writing – review & editing

Frederik Coppens Funding acquisition, Supervision, Writing – review & editing

Daniel Garijo Software, Writing – review and editing

Simone Leo Conceptualization, Project administration, Software, Writing – original draft

Marc Portier Writing – review & editing

Paul Groth Supervision

I am the main author of the corresponding manuscript and have contributed to all aspects of the research.

B.1.5 Contributions for *Formalizing RO-Crate in First Order Logic*

Section 4.3 on page 113 was published as an appendix in [Soiland-Reyes 2022a] (see B.1.3 on the preceding page).

I am the sole author of the corresponding appendix and have contributed to all aspects of the research.

B.1.6 Contributions for *Making Canonical Workflow Building Blocks interoperable across workflow languages*

Section 5.1 on page 121 was co-authored by:

Stian Soiland-Reyes Conceptualization, Funding acquisition, Investigation, Methodology, Project administration, Supervision, Writing – original draft, Writing – review & editing

Genís Bayarri Software, Software Documentation

Pau Andrio Methodology, Software, Validation, Software Documentation

Robin Long Software, Software Documentation

Douglas Lowe Software, Software Documentation

Ania Niewielska Methodology, Resources, Software

Adam Hospital Methodology, Project administration, Resuorces, Software, Validation, Visualization, Writing – original draft, Writing – review & editing

Paul Groth Methodology, Supervision, Writing – review & editing

I am the main author of the corresponding manuscript and have contributed to all aspects of the research.

B.1.7 Contributions for *The Specimen Data Refinery*

Section 5.2 on page 133 was co-authored by:

Alex Hardisty Conceptualization, Investigation, Supervision, Validation, Writing – original draft, Writing – review & editing, Approval.

Paul Brack Investigation, Writing – original draft, Writing – review & editing.

Carole Goble Conceptualization, Supervision, Writing – review & editing

Laurence Livermore Conceptualization, Funding acquisition, Investigation, Writing – original draft, Writing – review & editing.

Ben Scott Investigation, Writing – original draft, Writing – review & editing.

Quentin Groom Funding acquisition, Investigation, Writing – original draft, Writing – review & editing.

Stuart Owen Investigation, Writing – original draft, Writing – review & editing.

Stian Soiland-Reyes Investigation, Writing – original draft, Writing – review & editing.

My main contributions are to Section 5.2.2.2, 5.2.2.3, 5.2.4.1, 5.2.7. In the corresponding research I have contributed to designing, technical advice, insight and supervision.

B.1.8 Contributions for *Incrementally building FAIR Digital Objects with Specimen Data Refinery workflows*

Section 5.3 on page 148 was co-authored by:

Oliver Woolland Data curation, Resources, Software, Visualization, Writing – review & editing

Paul Brack Conceptualization, Software

Stian Soiland-Reyes Investigation, Methodology, Supervision, Writing – original draft, Writing – review & editing

Ben Scott Data curation, Software, Validation

Laurence Livermore Conceptualization, Data curation, Funding acquisition, Methodology, Project administration, Resources, Writing – review & editing

I am the main author of the corresponding manuscript and have contributed to all aspects of the research.

This work was presented as a poster by Stian Soiland-Reyes at First International Conference on FAIR Digital Objects, Leiden, The Netherlands.

- Poster: https://doi.org/10.5281/zenodo.7233688

B.1.9 Contributions for *Recording provenance of workflow runs with RO-Crate*

Section 5.4 on page 152 was co-authored by:

Simone Leo Conceptualization, Data Curation, Investigation, Methodology, Resources, Software, Supervision, Validation, Visualization, Writing – Original Draft preparation, Writing – Review & Editing

Michael R. Crusoe Conceptualization, Investigation, Software, Supervision

Laura Rodríguez-Navas Software, Writing – Original Draft preparation

Raül Sirvent Data Curation, Software, Writing – Original Draft preparation, Writing – Review & Editing

Alexander Kanitz Writing – Original Draft preparation, Writing – Review & Editing

Paul De Geest Data Curation, Software, Writing – Original Draft preparation

Rudolf Wittner Data Curation, Writing – Original Draft preparation, Writing – Review & Editing

Luca Pireddu Funding acquisition, Project Administration, Supervision, Writing – Review & Editing

Daniel Garijo Conceptualization, Formal Analysis, Writing – Original Draft preparation, Writing – Review & Editing

José M. Fernández Data Curation, Software, Writing – Original Draft preparation

Iacopo Colonnelli Data Curation, Software, Writing – Original Draft preparation

Matej Gallo Data Curation, Software

Tazro Ohta Data Curation, Software, Writing – Original Draft preparation

Hirotaka Suetake Data Curation, Software, Writing – Original Draft preparation

Salvador Capella-Gutierrez Funding Acquisition, Resources, Supervision, Writing – Original Draft preparation

Renske de Wit Software, Writing – Original Draft preparation, Writing – Review & Editing

Bruno de Paula Kinoshita Data Curation, Software, Writing – Original Draft preparation, Writing – Review & Editing

Stian Soiland-Reyes Conceptualization, Formal Analysis, Funding Acquisition, Investigation, Methodology, Resources, Software, Supervision, Visualization, Writing – Original Draft preparation, Writing – Review & Editing

I am the last author of this manuscript, and have contributed to all aspects of the research. My main contributions are in Sections 5.4.1, 5.4.5, 5.4.5.3, 5.4.5.4. I am supervising the Workflow Run Crate task force[2] together with its chairs Simone Leo and Laura Rodríguez-Navas.

B.1.10 Supplementary publications

I have also contributed as co-author to these articles during the PhD period, provided as supplements:

Supplement 1: *Ten Simple Rules for making a software tool workflow-ready*[3] [Brack 2022a]

Supplement 2: *Enhancing RDM in Galaxy by integrating RO-Crate*[4] [De Geest 2022]

Supplement 3: *Implementing FAIR Digital Objects in the EOSC-Life Workflow Collaboratory*[5] [Goble 2021]

Supplement 4: *Methods Included: Standardizing Computational Reuse and Portability with the Common Workflow Language*[6] [Crusoe 2022]

[2] https://www.researchobject.org/workflow-run-crate/#community

[3] https://s11.no/2022/phd/10-simple-rules-for-workflow-tools/

[4] https://s11.no/2022/phd/galaxy-ro-crate/

[5] https://s11.no/2021/phd/workflow-collaboratory/

[6] https://s11.no/2022/phd/methods-included/

Supplement 5: *Semantic micro-contributions with decentralised nanopublication services*[7] [Kuhn 2021]

Supplement 6: *Perspectives on automated composition of workflows in the life sciences*[8] [Lamprecht 2021]

Supplement 7: *ISO 23494: Biotechnology - Provenance Information Model for Biological Specimen and Data*[9] [Wittner 2020]

Supplement 8: *Toward a Common Standard for Data and Specimen Provenance in Life Sciences*[10] [Wittner 2023a]

Supplement 9: *A Community Roadmap for Scientific Workflows Research and Development*[11] [Ferreira da Silva 2021]

Supplement 10: *Unique, Persistent, Resolvable: Identifiers as the Foundation of FAIR*[12] [Juty 2020] (Main contribution pre-dates UvA affiliation)

Supplement 11: *FAIR Computational Workflows*[13] [Goble 2020] (*Main contribution pre-dates UvA affiliation*)

Supplement 12: *Sharing interoperable workflow provenance: A review of best practices and their practical application in CWLProv*[14] [Khan 2019] (Main contribution pre-dates UvA affiliation)

Supplement 13: *IEEE Standard for Bioinformatics Analyses Generated by High-Throughput Sequencing (HTS) to Facilitate Communication: IEEE Std 2791-2020*[15] [IEEE 2791-2020]

Supplement 14: *BioHackEU22 Project 22: Plant data exchange and standard interoperability*[16] [Arend 2022]

Supplement 15: *RO-Crate, a lightweight approach to Research Object data packaging*[17] [Ó Carragáin 2019b] (Main contribution pre-dates UvA affiliation)

Supplement 16: *Federated causal inference based on real-world observational data sources: application to a SARS-CoV-2 vaccine effectiveness assessment*[18] [Meurisse 2023]

Supplement 17: *Linking provenance and its metadata in multi-organizational environments of life sciences*[19] [Wittner 2023b]

[7] https://s11.no/2021/phd/nanopub/

[8] https://doi.org/10.12688/f1000research.54159.1

[9] https://s11.no/2021/phd/iso-23494-provenance/

[10] https://doi.org/10.1002/lrh2.10365

[11] https://doi.org/10.48550/arXiv.2110.02168

[12] https://doi.org/10.1162/dint_a_00025

[13] https://doi.org/10.1162/dint_a_00033

[14] https://doi.org/10.1093/gigascience/giz095

[15] https://research.manchester.ac.uk/en/publications/936de52b-ac53-4f0e-9927-77fd7073e88d

[16] https://doi.org/10.37044/osf.io/c724r

[17] https://s11.no/2019/phd/ro-crate/

[18] https://s11.no/2023/phd/federated-causal-inference/

[19] https://s11.no/2023/phd/linking-provenance/

Supplement 18: *BioHackEU22 Report: Enhancing Research Data Management in Galaxy and Data Stewardship Wizard by utilising RO-Crates*[20] [Eguinoa 2023]

Supplement 19: *BioHackEU23 report: Enabling continuous RDM using Annotated Research Contexts with RO-Crate profiles for ISA*[21] [Beier 2024]

Supplement 20: *BioHackEU23 report: Enabling FAIR Digital Objects with RO-Crate, Signposting and Bioschemas*[22] [Soiland-Reyes 2024a]

Supplement 21: *Report on "FAIR Signposting" and its uptake by the community*[23] [Wilkinson 2024]

Supplement 22: *Practical webby FDOs with RO-Crate and FAIR Signposting: Experiences and lessons learned*[24] [Soiland-Reyes 2024c]

I have been involved in All Aspects of the research for supplements 1, 2, 3, 4, 11, 12, 15, 18, 20, 22.

B.1.11 Contributor affiliations

Affiliations of co-authors (see Section B.1 on page 228), excluding supplements:

Pau Andrio **https://orcid.org/0000-0003-2116-3880** The Spanish National Bioinformatics Institute (INB), Barcelona
Supercomputing Center (BSC), Barcelona, Spain

Genís Bayarri **https://orcid.org/0000-0003-0513-0288** Institute for Research in Biomedicine (IRB Barcelona), The Barcelona Institute of Science and Technology (BIST), Barcelona, Spain

Paul Brack **https://orcid.org/0000-0002-5432-2748** Department of Computer Science, The University of Manchester, Manchester, UK (former)

Eoghan Ó Carragáin **https://orcid.org/0000-0001-8131-2150** University College Cork, Ireland

Iacopo Colonnelli **https://orcid.org/0000-0001-9290-2017** Computer Science Dept., Università degli Studi di Torino, Torino, Italy

Frederik Coppens **https://orcid.org/0000-0001-6565-5145** Department of Plant Biotechnology and Bioinformatics, Ghent University, Ghent, Belgium
VIB-UGent Center for Plant Systems Biology, Ghent, Belgium

Mercè Crosas **https://orcid.org/0000-0003-1304-1939** Barcelona Supercomputing Center (BSC), Barcelona, Spain
The Committee on Data of the International Science Council (ISC) (CODATA)
Secretària de Govern Obert, Catalunya, Barcelona, Spain (former)

[20] https://s11.no/2023/phd/enhancing-rdm-galaxy-dsw/
[21] https://doi.org/10.37044/osf.io/7y2jh
[22] https://s11.no/2024/enabling-fair-digital-objects/
[23] https://s11.no/2024/signposting-report/
[24] https://s11.no/2024/webby-fdos/

Institute for Quantitative Social Science, Harvard University, Cambridge, MA, USA (former)

Michael R. Crusoe https://orcid.org/0000-0002-2961-9670 Department of Computer Science, Vrije Universiteit Amsterdam, Amsterdam, The Netherlands
DTL Projects, The Netherlands
Forschungszentrum Jülich, Jülich, Germany
Common Workflow Language project, Software Freedom Conservancy, Brooklyn, NY, USA

Matej Gallo https://orcid.org/0000-0002-1119-1792 Faculty of Informatics, Masaryk University, Brno, Czech Republic

Paul De Geest https://orcid.org/0000-0002-8940-4946 VIB-UGent Center for Plant Systems Biology, Ghent, Belgium

Ignacio Eguinoa https://orcid.org/0000-0002-6190-122X Showpad, Ghent, Belgium
Department of Plant Biotechnology and Bioinformatics, Ghent University, Ghent, Belgium (former)
VIB-UGent Center for Plant Systems Biology, Ghent, Belgium (former)

José Mª Fernández https://orcid.org/0000-0002-4806-5140 Barcelona Supercomputing Center, Barcelona, Spain

Daniel Garijo https://orcid.org/0000-0003-0454-7145 Ontology Engineering Group, Universidad Politécnica de Madrid, Madrid, Spain

Carole Goble https://orcid.org/0000-0003-1219-2137 Department of Computer Science, The University of Manchester, Manchester, UK

Quentin Groom https://orcid.org/0000-0002-0596-5376 Meise Botanic Garden, Meise, Belgium

Paul Groth https://orcid.org/0000-0003-0183-6910 Informatics Institute, University of Amsterdam, Amsterdam, The Netherlands

Björn Grüning https://orcid.org/0000-0002-3079-6586 Bioinformatics Group, Department of Computer Science, Albert-Ludwigs-University Freiburg, Freiburg, Germany

Alex Hardisty https://orcid.org/0000-0002-0767-4310 School of Computer Science and Informatics, Cardiff University, Cardiff, UK (former)

Adam Hospital https://orcid.org/0000-0002-8291-8071 Institute for Research in Biomedicine (IRB Barcelona), The Barcelona Institute of Science and Technology (BIST), Barcelona, Spain

Leyla Jael Castro https://orcid.org/0000-0003-3986-0510 ZB MED Information Centre for Life Sciences, Cologne, Germany

Simone Leo https://orcid.org/0000-0001-8271-5429 Center for Advanced Studies, Research, and Development in Sardinia (CRS4), Pula (CA), Italy

Alexander Kanitz https://orcid.org/0000-0002-3468-0652 Biozentrum, University of Basel, Basel, Switzerland Swiss Institute of Bioinformatics, Lausanne, Switzerland

Bruno de Paula Kinoshita https://orcid.org/0000-0001-8250-4074 Barcelona Supercomputing Center (BSC), Barcelona, Spain

Laurence Livermore https://orcid.org/0000-0002-7341-1842 The Natural History Museum, London, UK

Robin Long https://orcid.org/0000-0003-2249-645X Lancaster University, Lancaster, UK Research IT, The University of Manchester, Manchester, UK (former)

Douglas Lowe https://orcid.org/0000-0002-1248-3594 Research IT, The University of Manchester, Manchester, UK

Ania Niewielska https://orcid.org/0000-0003-0989-3389 European Bioinformatics Institute (EMBL-EBI), Cambridge, UK

Tazro Ohta https://orcid.org/0000-0003-3777-5945 Database Center for Life Science, Joint Support-Center for Data Science Research, Research Organization of Information and Systems, Shizuoka, Japan Institute for Advanced Academic Research, Chiba University, Chiba, Japan

Stuart Owen https://orcid.org/0000-0003-2130-0865 Department of Computer Science, The University of Manchester, Manchester, UK

Luca Pireddu https://orcid.org/0000-0002-4663-5613 Center for Advanced Studies, Research and Development in Sardinia (CRS4), Pula, Italy

Marc Portier https://orcid.org/0000-0002-9648-6484 Vlaams Instituut voor de Zee (VLIZ), Oostende, Belgium

Laura Rodriguez-Navas https://orcid.org/0000-0003-4929-1219 Universitat Oberta de Catalunya (UOC), Barcelona, Spain
Life Sciences Department. Barcelona Supercomputing Center (BSC), Barcelona, Spain (former)

Marco La Rosa https://orcid.org/0000-0001-5383-6993 PARADISEC, Melbourne, Australia

Ben Scott https://orcid.org/0000-0002-5590-7174 The Natural History Museum, London, UK

Peter Sefton https://orcid.org/0000-0002-3545-944X School of Languages and Cultures, The University of Queensland, Brisbane, Queensland, Australia
Faculty of Science, University Technology Sydney, Australia (former)

Raül Sirvent https://orcid.org/0000-0003-0606-2512 Barcelona Supercomputing Center (BSC), Barcelona, Spain

Stian Soiland-Reyes **https://orcid.org/0000-0001-9842-9718** Department of Computer Science, The University of Manchester, Manchester, UK
Informatics Institute, University of Amsterdam, Amsterdam, The Netherlands

Hirotaka Suetake **https://orcid.org/0000-0003-2765-0049** Department of Creative Informatics, Graduate School of Information Science and Technology, The University of Tokyo, Tokyo, Japan

Ana Trisovic **https://orcid.org/0000-0003-1991-0533** Institute for Quantitative Social Science, Harvard University, Cambridge, MA, USA

Alan R Williams **https://orcid.org/0000-0003-3156-2105** Department of Computer Science, The University of Manchester, Manchester, UK (former)

Renske de Wit **https://orcid.org/0000-0003-0902-0086** Vrije Universiteit Amsterdam, Amsterdam, The Netherlands

Rudolf Wittner **https://orcid.org/0000-0002-0003-2024** Faculty of Informatics, Masaryk University, Brno, Czech Republic Institute of Computer Science, Masaryk University, Brno, Czech Republic BBMRI-ERIC, Neue Stiftingtalstrasse 2, 8010, Graz, Austria

Oliver Woolland **https://orcid.org/0000-0002-4565-9760** Research IT, The University of Manchester, Manchester, UK

B.2 Community roles

For Chapter 3 and Section 5.1 I am a member of FAIR Digital Object Forum [FDO] working groups FDO-CWFR, FDO-SEM, FDO-TSIG2 and have contributions to FDO specifications [Anders 2023a, Anders 2023b], and to the FDO demonstrator paper [Wittenburg 2022a]. I am a member of the FDO 2024 programme committee[25] and the Research Data Alliance (RDA's) FAIR Digital Object Fabric[26] Interest Group.

For Chapter 4 I co-chair the RO-Crate community[27][28] together with Peter Sefton. We are the main editors and authors of the RO-Crate specifications [RO-Crate 1.0, RO-Crate 1.1, RO-Crate 1.1.3, RO-Crate 1.2].

For Section 5.1 I was deputy work package leader in BioExcel-2, with Adam Hospital as work package leader. I am a member of the BioExcel-3 Scientific Advisory Board.

For Section 5.4 I am a member of the Workflow Run RO-Crate[29] community as well as the Workflows Community Initiative[30] working group FAIR Computational Workflow[31]

[25] https://fairdo.org/fdo2024-conference/
[26] https://www.rd-alliance.org/group/FAIR-digital-object-fabric-ig.html
[27] https://www.researchobject.org/ro-crate/community
[28] see Section A.2.1 on page 213
[29] https://www.researchobject.org/workflow-run-crate/#community
[30] https://workflows.community/
[31] https://workflows.community/groups/fair/

For Supplement 4 [Crusoe 2022] and 12 [Khan 2019] I am a member of the Common Workflow Language leadership team[32].

For Supplement 13 [IEEE 2791-2020] I was a member of the BioCompute Object[33] technical steering committee and a member of the IEEE 2791-2020 working group.

B.3 Software contributions

During this PhD I have contributed to several software applications and libraries, including:

- signposting[34] [Soiland-Reyes 2022e], link parser library for Python (main author)
- Benchmarks for Apples-to-Apples FAIR Signposting[35], main author and maintainer (see [Wilkinson 2022a, Wilkinson 2024])
- ro-crate-py[36] [De Geest 2023a] (initial author, contributor; main author is Simone Leo)
- ro-index-paper[37] – early prototype for survey of Research Object usage
- runcrate[38] [Leo 2023a] contributor, main author is Simone Leo
- ro-crate-preview[39], GitHub action to build HTML preview of RO-Crate. Contributed as supervisor, documentation, bug fixes. Main author is Gerard Capes.
- cwlviewer[40] [Robinson 2023], contributed feature[41], main author is Mark Robinson (see [Robinson 2017])
- ro-crate-validator-py[42], supervisor, main author is Xuanqi “Logan” Li

B.4 Standard contributions

- RO-Crate Specification 1.1.3 [RO-Crate 1.1.3], contributing as co-chair of RO-Crate community and editor.
- RO-Crate Specification 1.2-DRAFT[43] [RO-Crate 1.2]. I am the main editor of this planned release and have contributed several new sections including RO-Crate profiles[44]

[32] https://www.commonwl.org/governance/
[33] https://www.biocomputeobject.org/
[34] https://pypi.org/project/signposting/
[35] https://w3id.org/a2a-fair-metrics/
[36] https://pypi.org/project/rocrate/
[37] https://github.com/stain/ro-index-paper
[38] https://github.com/ResearchObject/runcrate
[39] https://github.com/marketplace/actions/ro-crate-preview
[40] https://view.commonwl.org/
[41] https://github.com/common-workflow-language/cwlviewer/pull/241
[42] https://github.com/ResearchObject/ro-crate-validator-py
[43] https://www.researchobject.org/ro-crate/1.2-DRAFT/
[44] https://www.researchobject.org/ro-crate/1.2-DRAFT/profiles

- FAIR digital object technical overview [Anders 2023b], contributing clarifications.
- FDO requirement specifications [Anders 2023a], contributing as member of the FDO TSIG group.
- IEEE 2791-2020 [IEEE 2791-2020], contributing as member of P2791 Working Group. I was responsible for aspects of identifiers and internal review.
- JSON Schema for IEEE 2791[45], contributing as member of P2791 Working Group and internal review.
- RFC9264 Linkset [Wilde 2020], contributed[46] JSON-LD context and reviewed.
- ISO/TS 23494-1:2023 & ISO/AWI 23494-2, contributed as consultant to ISO/TC 276 provenance group (see [Wittner 2020, Wittner 2023a, Wittner 2023b])

B.4.1 RO-Crate profiles

- Workflow RO-Crate Profile 1.0 [Bacall 2022]
  https://w3id.org/workflowhub/workflow-ro-crate/1.0
- Common Provenance Model RO-Crate profile 0.2
  https://w3id.org/cpm/ro-crate/0.2
- Five Safes RO-Crate profile 0.4 [Soiland-Reyes 2023d]
  https://w3id.org/5s-crate/0.4
- Process Run Crate specification 0.4 [WRROC 2023a]
  https://w3id.org/ro/wfrun/process/0.4
- Workflow Run Crate specification 0.4 [WRROC 2023b]
  https://w3id.org/ro/wfrun/workflow/0.4
- Provenance Run Crate specification 0.4 [WRROC 2023c]
  https://w3id.org/ro/wfrun/provenance/0.4

B.5 Training material and training events

- Leyla Jael Castro, Stian Soiland-Reyes, Jonas Grieb, Claus Weiland (2024):
  Practical web-based FDOs with RO-Crate and FAIR Signposting.
  International FAIR Digital Objects Implementation Summit 2024, Berlin, Germany, 2024-03-20/–21.
  https://doi.org/10.5281/zenodo.10892090
- Stian Soiland-Reyes, Claus Weiland, Herbert Van de Sompel, Leyla Jael Castro (2024):
  Improving FAIRability of your research outcomes with RO-Crates, SignPosting and

[45] https://w3id.org/ieee/ieee-2791-schema
[46] https://github.com/dret/I-D/pull/129

Bioschemas.
15th International SWAT4HCLS Conference, Leiden, The Netherlands, 2024-26-26/–29

- **Packaging Data using RO-Crate**.
  Galaxy Smörgåsbord 2023
  International FAIR Digital Objects Implementation Summit 2024, Berlin, Germany, 2024-03-20/–21. (main author: Douglas Lowe)
  http://docs.bioexcel.eu/cwl-best-practice-guide/

- **Common Workflow Language Engines**.
  (main author: Robin Long)
  http://docs.bioexcel.eu/cwl-engine-guide/

B.6 Dataset contributions

- Zenodo metadata JSON records as of 2019-09-16
  https://doi.org/10.5281/zenodo.3531504

- Open PHACTS Linksets 2.1.1
  https://doi.org/10.5281/zenodo.4704867

- RO-Crate of RO-Crate specification 1.1 [RO-Crate 1.1.3]
  https://www.researchobject.org/ro-crate/1.1/ro-crate-preview.html

- RO-Crate of RO-Crate specification 1.2-DRAFT [RO-Crate 1.2]
  https://www.researchobject.org/ro-crate/1.2-DRAFT/ro-crate-preview.html

- Packaging research artefacts with RO-Crate [Soiland-Reyes 2022g]
  https://w3id.org/ro/doi/10.5281/zenodo.5146227

- Comparison tables for evaluating FAIR Digital Object and Linked Data [Soiland-Reyes 2023a]
  https://w3id.org/ro/doi/10.5281/zenodo.8075229

- BY-COVID WP5 T5.2 Baseline Use Case
  https://w3id.org/ro/doi/10.5281/zenodo.6913045

- Linking provenance and its metadata for an AI-based computation using CPM and RO-Crate
  https://doi.org/10.5281/zenodo.10245846

- Packing provenance using CPM RO-Crate profile [Wittner 2023c]
  https://doi.org/10.5281/zenodo.8095888

- Recording provenance of workflow runs with RO-Crate (RO-Crate and mapping) [Leo 2023b]
  https://w3id.org/ro/doi/10.5281/zenodo.10368989

B.7 Presentation contributions

Stian Soiland-Reyes, Herbert van de Sompel (2024):
Signposting and RO-Crate: experiences and lessons learned.
International FAIR Digital Objects Implementation Summit 2024, Berlin, Germany, 2024-03-20/–21.
https://doi.org/10.5281/zenodo.10847062

Stian Soiland-Reyes, Leyla Jael Garcia (2023):
Overview of FAIR data publishing with Bioschemas & RO-Crate.
ELIXIR All Hands meeting 2023, workshop "Building lightweight FAIR data packages with Bioschemas and RO-Crate", Dublin, Ireland, 2023-06-05/–08
https://doi.org/10.7490/f1000research.1119459.1

Stian Soiland-Reyes, Carole Goble (2023):
Building diverse collections using RO-Crate.
ELIXIR All Hands meeting 2023, mini-symposium "Biodiversity, Food Security and Pathogens", Dublin, Ireland, 2023-06-05/–08
(presented by Stian Soiland-Reyes)
https://doi.org/10.7490/f1000research.1119466.1

Stian Soiland-Reyes, Carole Goble (2023):
Building diverse FDO Collections using RO-Crate.
FAIR Digital Object Forum, workshop "Defining FDO Collections", 2023-04-14.
(presented by Stian Soiland-Reyes)
https://doi.org/10.5281/zenodo.7828632
https://youtu.be/5GYdN5B1tc8

Stian Soiland-Reyes, Herbert Van De Sompel (2023):
Enabling FAIR Signposting and RO-Crate for content/metadata discovery and consumption.
FAIR-IMPACT Open Call for Support (Webinar), 2023-03-27
https://doi.org/10.5281/zenodo.7774582

Carole Goble, Stian Soiland-Reyes (2023):
Sharing research artefacts as FAIR Digital Objects using RO-Crate.
Brookhaven National Laboratory, 2023-01-23.
(presented by Stian Soiland-Reyes)
https://doi.org/10.5281/zenodo.7559338
https://youtu.be/0T4FBbpgtQo

Justin Clark-Casey, Stian Soiland-Reyes (2022):
Making EOSC Research Objects FAIR with RO-Crate: A common metadata overlay for EOSC repositories.
EOSC Symposium 2022
(presented by Justin Clark-Casey)
https://doi.org/10.5281/zenodo.7323480

Stian Soiland-Reyes, Leyla Jael Castro, Daniel Garijo, Marc Portier, Carole Goble, Paul Groth (2022):
Updating Linked Data practices for FAIR Digital Object principles.
1st International Conference on FAIR Digital Objects (FDO 2022) (presented by Stian Soiland-Reyes)
https://doi.org/10.5281/zenodo.7256428

Stian Soiland-Reyes (2021):
RO-Crate — A brief "crash course".
ELIXIR Data-Interoperability Joint Platform F2F Hybrid Meeting, 2021-11-23.
https://slides.com/soilandreyes/2021-11-23-ro-crate-crash-course/

Stian Soiland-Reyes (2021):
Reproducibility; Research Objects (RO-Crate) and Common Workflow Language (CWL).
WoSSS21:Workshop on Sustainable Software Sustainability, 2021-10-07.
https://slides.com/soilandreyes/2021-10-07-reproducibility-research-objects
https://www.youtube.com/watch?v=vNHqTcHnfyI

Stian Soiland-Reyes (2021):
Sharing FAIR Research Objects to improve reproducibility.
ZB-Med Seminar, 2021-07-15.
[video recording] https://doi.org/10.5281/zenodo.5105857

Stian Soiland-Reyes (2021):
RO-Crate, workflows and FAIR Digital Objects.
FAIR Digital Object Forum, CWFR & FDO SEM meeting, 2021-07-02
https://youtu.be/gTT0m_zQsPU
http://slides.com/soilandreyes/2021-07-02-ro-crate-workflows-fdo
https://doi.org/10.5281/zenodo.5060283

Stian Soiland-Reyes (2021):
Capturing "Just enough" Data, Software and Metadata with RO-Crate.
FAIR Festival 2021, FAIR Implementation Challenges & Solutions, 2021-06-21.
http://slides.com/soilandreyes/2021-06-21-capturing-just-enough-data-software-and-metadata-with-ro-crate
https://doi.org/10.5281/zenodo.5007432

Stian Soiland-Reyes (2021):
Capturing Just Enough Data, Software and Metadata with RO-Crate.
Dataverse community meeting 2021, Software Metadata and Containerization, 2021-06-17.
http://slides.com/soilandreyes/2021-06-17-capturing-just-enough-with-ro-crate
https://youtu.be/LJq-mzT9v8o?t=1731
https://doi.org/10.5281/zenodo.4973678

Stian Soiland-Reyes (2021): **Capturing workflow life cycle with RO-Crate**.

ELIXIR All Hands 2021, Workshop: Workflow Life Cycle, 2021-06-11
http://slides.com/soilandreyes/2021-06-11-ro-crate-workflows https://doi.org/10.5281/zenodo.4926088

Stian Soiland-Reyes (2021):
Describing and packaging workflows using RO-Crate and BioCompute Objects.
Webinar for U.S. Food and Drug Administration (FDA), 2021-05-12. https://youtu.be/3APqPwRIRkA
https://doi.org/10.5281/zenodo.4633732

Stian Soiland-Reyes (2021):
Data provenance with RO-Crate.
EOSC-Life retreat 2021, Provenance of tools and workflows; FAIRification of workflows, 2021-05-19.
http://slides.com/soilandreyes/2021-05-19-recording-provenance-with-ro-crate

Stian Soiland-Reyes, Carole Goble (2021):
RO-Crate: Describing and packaging FAIR Research Objects.
Scottish Covid-19 Response Consortium, 2021-03-18 https://doi.org/10.5281/zenodo.4633655

Stian Soiland-Reyes, Carole Goble (2021):
Publishing workflows in WorkflowHub.eu using CWL, and packaging with RO-Crate.
2021 Common Workflow Language Virtual Conference
https://youtu.be/_tyMPj4emw0

Stian Soiland-Reyes (2020): **Packaging workflows with RO-Crate** *FAIR Workflows* workshop at International FAIR Convergence Symposium, 2020-11-30. Video recording Slides

Stian Soiland-Reyes, Ignacio Eguinoa (2020):
Packaging workflows with RO-Crate.
Workshop on FAIR Computational Workflows, 19th European Conference on Computational Biology (ECCB 2020).
https://doi.org/10.5281/zenodo.4011999

Eoghan Ó Carragáin, Carole Goble, Peter Sefton, Stian Soiland-Reyes (2019):
RO-Crate, a lightweight approach to Research Object data packaging..
RO-15 at *Workshop on Research Objects (RO 2019)*, IEEE eScience 2019, 2019-09-24, San Diego, CA, USA.
(presented by Stian Soiland-Reyes)
http://slides.com/soilandreyes/2019-09-24-ro-crate
https://doi.org/10.5281/zenodo.3337883

Eoghan Ó Carragáin, Carole Goble, Peter Sefton, Stian Soiland-Reyes (2019):
RO-Crate, a lightweight approach to Research Object data packaging.
Talk at *Bioinformatics Open Source Conference (BOSC2019)*. (presented by Stian Soiland-Reyes)
F1000Research 2019, **8**(ISCB Comm J):1196 (slides)

https://slides.com/soilandreyes/2019-07-24-bosc-ro-crate
https://doi.org/10.7490/f1000research.1117129.1

B.8 Workshop organizing

Carole Goble, Raul Palma, Stian Soiland-Reyes, Daniel Garijo (2019):
Workshop on Research Objects 2019 (RO2019). *eScience 2019*, San Diego, California, US, 2019-09-24.
http://www.researchobject.org/ro2019/

Ignacio Eguinoa, Carole Goble, Michael R Crusoe, Stian Soiland-Reyes, Salvador Capella-Gutierrez, Sarah Cohen-Boulakia, Björn Grüning, Alexander Peltzer, Simone Leo, Frederik Coppens, Mateusz Kuzak (2020):
Workshop on FAIR Computational Workflows.
19th European Conference on Computational Biology (ECCB 2020).

Carole Goble, Stian Soiland-Reyes, Salvador Capella-Gutierrez, José Mª Fernández, Frederik Coppens (2021):
Workflow Life Cycle workshop.
ELIXIR All Hands 2021 (virtual) 2021-06-11. https://workflowhub.eu/events/4

Stian Soiland-Reyes, Leyla Jael Castro, Núria Queralt Rosinach (2023):
Building lightweight FAIR data packages with Bioschemas and RO-Crate.
Workshop at *ELIXIR All Hands meeting 2023* (AHM2023), Dublin, Ireland, 2023-06-06
https://elixir-events.eventscase.com/EN/elixirallhands2023/Agenda https://docs.google.com/document/d/1Vh9mUBWvNEsvC5YZRITtZxsyE6Wr18rJUgp9KweJFNg

Tom Giles, Stian Soiland-Reyes, Jonathan Couldridge (2023):
Approaching Five Safes with TRE-FX Trusted Workflow Run Crate.
TRE-FX Virtual Stakeholder Workshop, 2023-07-11
https://trefx.uk/2023-07-11-tre-stakeholder-workshop

Stian Soiland-Reyes, Leyla Jael Castro, Dietrich Rebholz-Schuhmann (2023):
Data exchange with RO-Crates and Knowledge Graphs.
Workshop at *Open Science Festival 2023*, Cologne, Germany, 2023-07-05.
https://www.zbmed.de/vernetzen/veranstaltungen/open-science-festival/data-exchange-with-ro-crates-and-knowledge-graphs https://tinyurl.com/osfcrate

B.9 Poster contributions

- Eoghan Ó Carragáin, Carole Goble, Peter Sefton, Stian Soiland-Reyes (2019):
  RO-Crate, a lightweight approach to Research Object data packaging [version 1; not peer reviewed].
  Poster at *Bioinformatics Open Source Conference (BOSC2019)*. *F1000Research* 2019, **8**(ISCB

Comm J):1197 (poster)
https://doi.org/10.7490/f1000research.1117130.1

- **RO-Crate, a lightweight approach to Research Object data packaging**.
  Bioinformatics Open Source Conference (BOSC) (BOSC), ISMB/ECCB 2019, Basel, Switzerland, 24-25 July 2019
  https://doi.org/10.5281/zenodo.3343031
  https://doi.org/10.7490/f1000research.1117130.1

- **ISO 23494: Biotechnology - Provenance Information Model for Biological Specimen and Data**.
  Provenance Week 2020
  (presented by Rudolf Wittner)
  https://doi.org/10.5281/zenodo.5004842

- **Improving Galaxy provenance export using RO-Crate**.
  1st International Conference on FAIR Digital Objects (FDO2022), Leiden, The Netherlands, 2022-10-26/–28
  https://doi.org/10.5281/zenodo.7257146

- **Creating lightweight FAIR Digital Objects with RO-Crate and FAIR Signposting**.
  1st International Conference on FAIR Digital Objects (FDO2022), Leiden, The Netherlands, 2022-10-26/–28 https://doi.org/10.5281/zenodo.7245315

- **Incrementally building FAIR Digital Objects with Specimen Data Refinery workflows**.
  1st International Conference on FAIR Digital Objects (FDO2022), Leiden, The Netherlands, 2022-10-26/–28
  (presented by Paul De Geest)
  https://doi.org/10.5281/zenodo.7233688

- **Making workflow provenance FAIR across workflow systems with Workflow Run RO-Crate**.
  ELIXIR All Hands 2023, Dublin, Ireland, 2023-06-05 / –08
  https://doi.org/10.5281/zenodo.8004793
  https://doi.org/10.7490/f1000research.1119445.1

- **Sharing data as machine-actionable objects using RO-Crate, Bioschemas and Signposting**.
  ELIXIR All Hands 2023, Dublin, Ireland, 2023-06-05 / –08
  https://doi.org/10.5281/zenodo.8004796

- **WorkflowHub – a fair registry for workflows** *ELIXIR All Hands 2023*, Dublin, Ireland, 2023-06-05 / –08
  (presented by Carole Goble)
  https://doi.org/10.7490/f1000research.1119430.1

Bibliography

[van der Aalst 2014] Wil M. P. van der Aalst (2014):
Data Scientist: The Engineer of the Future.
Proceedings of the I-ESA Conferences **7** (IESACONF)
https://doi.org/10.1007/978-3-319-04948-9_2

[Addink 2019] Wouter Addink, Dimitrios Koureas, Ana Rubio (2019):
DiSSCo as a New Regional Model for Scientific Collections in Europe.
Biodiversity Information Science and Standards **3**:e37502.
https://doi.org/10.3897/biss.3.37502

[Afgan 2018] Enis Afgan, Dannon Baker, Bérénice Batut, Marius van den Beek, Dave Bouvier, Martin Čech, John Chilton, Dave Clements, Nate Coraor, Björn A Grüning, Aysam Guerler, Jennifer Hillman-Jackson, Saskia Hiltemann, Vahid Jalili, Helena Rasche, Nicola Soranzo, Jeremy Goecks, James Taylor, Anton Nekrutenko, Daniel Blankenberg (2018):
The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2018 update.
Nucleic Acids Research **46**(W1) W537–W544
https://doi.org/10.1093/nar/gky379

[Afgan 2023] Enis Afgan, Istvan Albert, Renato Alves et al. (2023):
galaxyproject/galaxy version 23.1.1
GitHub
https://github.com/galaxyproject/galaxy/releases/tag/v23.1.1
https://identifiers.org/swh:1:rel:33ce0ce4f6e3d77d5c0af8cff24b2f68ba8d57e9

[Agarwal 2021] Deborah Agarwal, Carole Goble, Stian Soiland-Reyes, Ugis Sarkans, Daniel Noesgaard, Uwe Schindler, Martin Fenner, Paolo Manghi, Shelley Stall, Caroline Coward, Chris Erdmann (2021):
Data Citation Community of Practice – 8 June 2021 Workshop.
Zenodo/AGU
https://data.agu.org/DataCitationCoP/2nd-workshop-data-citation
https://doi.org/10.5281/zenodo.4916734

[Albertoni 2020] Riccardo Albertoni, David Browning, Simon Cox, Alejandra Gonzalez Beltran, Andrea Perego, Peter Winstanley, Dataset Exchange Working Group (2020):
Data Catalog Vocabulary (DCAT) – Version 2.
W3C Recommendation (2020) 04 February 2020
https://www.w3.org/TR/2020/REC-vocab-dcat-2-20200204/

[Albertoni 2024] Riccardo Albertonim, David Browning, Simon Cox, Alejandra Gonzalez Beltran, Andrea Perego, Peter Winstanley (2024):
Data Catalog Vocabulary (DCAT)- Version 3.
W3C Candidate Recommendation 18 January 2024
https://www.w3.org/TR/2024/CR-vocab-dcat-3-20240118/

[Allan 2019] E Louise Allan, Laurence Livermore, Benjamin Price, Olha Shchedrina, Vincent Smith (2019):
A Novel Automated Mass Digitisation Workflow for Natural History Microscope Slides.
Biodiversity Data Journal **7**:e32342.
https://doi.org/10.3897/BDJ.7.e32342

[Allcock 2005] William Allcock, John Bresnahan, Rajkumar Kettimuthu, Michael Link, Catalin Dumitrescu, Ioan Raicu, Ian Foster (2005):
The Globus Striped GridFTP Framework and Server.
SC '05: Proceedings of the 2005 ACM/IEEE Conference on Supercomputing, Seattle, WA, USA, IEEE
https://doi.org/10.1109/sc.2005.72
https://marketing.globuscs.info/production/strapi/uploads/gridftp_final_cca95d9e12.pdf

[Almeida 2019] Alexandre Almeida, Alex L. Mitchell, Miguel Boland, Samuel C. Forster, Gregory B. Gloor, Aleksandra Tarkowska, Trevor D. Lawley, Robert D. Finn (2019):
A new genomic blueprint of the human gut microbiota.
Nature **568**(7753) 499–504.
https://doi.org/10.1038/s41586-019-0965-1

[Alterovitz 2018] Gil Alterovitz, Dennis A Dean II, Carole Goble, Michael R Crusoe, Stian Soiland-Reyes, Amanda Bell, Anais Hayes, Anita Suresh, Charles Hadley S King IV, Dan Taylor, KanakaDurga Addepalli, Elaine Johanson, Elaine E Thompson, Eric Donaldson, Hiroki Morizono, Hsinyi Tsang, Jeet K Vora, Jeremy Goecks, Jianchao Yao, Jonas S Almeida, Jonathon Keeney, KanakaDurga Addepalli, Konstantinos Krampis, Krista Smith, Lydia Guo, Mark Walderhaug, Marco Schito, Matthew Ezewudo, Nuria Guimera, Paul Walsh, Robel Kahsay, Srikanth Gottipati, Timothy C Rodwell, Toby Bloom, Yuching Lai, Vahan Simonyan, Raja Mazumder (2018):
Enabling precision medicine via standard communication of HTS provenance, analysis, and results.
PLOS Biology **16**(12):e3000099
https://doi.org/10.1371/journal.pbio.3000099

[Alves 2021] Renato Alves, Dimitrios Bampalikis, Leyla Jael Castro, José María Fernández, Jennifer Harrow, Mateusz Kuzak, Eva Martin, Fotis E. Psomopoulos, Allegra Via (2021):
ELIXIR Software Management Plan for Life Sciences.
BioHackrXiv
https://doi.org/10.37044/osf.io/k8znb

[Amorim 2016] Ricardo Carvalho Amorim, João Aguiar Castro, João Rocha da Silva, Cristina Ribeiro (2016):
A comparison of research data management platforms: Architecture, flexible metadata and interoperability.
Universal Access in the Information Society **16** pp 851–862.
https://doi.org/10.1007/s10209-016-0475-y

[Amstutz 2021] Peter Amstutz, Maxim Mikheev, Michael R. Crusoe, Nebojša Tijanić, Samuel Lampa et al. (2022):
Existing Workflow systems.
Common Workflow Language wiki, GitHub. https://s.apache.org/existing-workflow-systems updated 2023-09-09, accessed 2023-11-09.

[Amstutz 2023] Peter Amstutz, Michael R. Crusoe, Farah Zaib Khan, Stian Soiland-Reyes, Manvendra Singh, Kapil kumar, John Chilton, Thomas Hickman, boysha, Tomoya Tanjo, Rupert Nash, Kevin Hannon, ash, Michael Kotliar, Brad Chapman, Andrey Kartashov, Guillermo Carrasco, Dan Leehr, Nebojsa Tijanic, Joshua C. Randall, Miguel Boland, bogdang989, Chuck McCallum, Hervé Ménager, Pau Ruiz Safont, Bruno P. Kinoshita, Denis Yuen, Gijs Molenaar (2023):
common-workflow-language/cwltool: 3.1.20230127121939.
https://doi.org/10.5281/zenodo.7575947

[Anders 2022] Ivonne Anders, Maggie Hellström, Sharif Islam, Thomas Jejkal, Larry Lannom, Ulrich Schwardmann, Peter Wittenburg (2022):
FDO PID profiles & attributes
FDO Specification Documents PR-PIDProfileAttributes-2.1-20221017
FAIR Digital Objects Forum
https://doi.org/10.5281/zenodo.7825630

[Anders 2023a] Ivonne Anders, Christophe Blanchi, Daan Broder, Maggie Hellström, Sharif Islam, Thomas Jejkal, Larry Lannom, Karsten Peters-von Gehlen, Robert Quick, Alexander Schlemmer, Ulrich Schwardmann, Stian Soiland-Reyes, George Strawn, Dieter van Uytvanck, Claus Weiland, Peter Wittenburg, Carlo Zwölf (2023):
FAIR Digital Objects Forum FDO requirement specifications. Version 3.0.
George Strawn, Peter Wittenburg (eds.)
FDO Specification Documents PR-RequirementSpec-3.0
FAIR Digital Objects Forum
https://doi.org/10.5281/zenodo.7782262

[Anders 2023b] Ivonne Anders, Christophe Blanchi, Daan Broder, Maggie Hellström, Sharif Islam, Thomas Jejkal, Larry Lannom Larry, Karsten Peters-von Gehlen, Robert Quick, Alexander Schlemmer, Ulrich Schwardmann, Stian Soiland-Reyes, George Strawn, Dieter van Uytvanck, Claus Weiland, Peter Wittenburg, Carlo Zwölf (2023):
FAIR digital object technical overview. Version PEN 2.0.
FDO Specification Documents Full FDO Overview PEN-2.0-v2
FAIR Digital Objects Forum
https://doi.org/10.5281/zenodo.7824714

[Andrio 2019] Pau Andrio, Adam Hospital, Javier Conejero, Luis Jordá, Marc Del Pino, Laia Codo, Stian Soiland-Reyes, Carole Goble, Daniele Lezzi, Rosa M. Badia, Modesto Orozco, Josep Ll. Gelpi (2019):
BioExcel Building Blocks, a software library for interoperable biomolecular simulation workflows.
Scientific Data **6**:169
https://doi.org/10.1038/s41597-019-0177-4

[ANSI/NISO Z39.99-2017] NISO (2017):
ANSI/NISO Z39.99-2017, ResourceSync Framework Specification.
National Information Standards Organization ResourceSync Standing Committee
https://doi.org/10.3789/ansi.niso.z39.99-2017
http://www.openarchives.org/rs/1.1/resourcesync

[Arend 2022] Daniel Arend, Sebastian Beier, laurent bouri, Marco Brandizi, Donald Hobern, Erwan Le Floch, Timo Mühlhaus, Cyril Pommier, Stuart Owen, Philippe Rocca-Serra, Thomas Rosnet, Stian Soiland-Reyes (2022):
BioHackEU22 Project 22: Plant data exchange and standard interoperability.
BioHackrXiv
https://doi.org/10.37044/osf.io/c724r

[Arfaoui 2020] Ghaith Arfaoui, Maroua Jaoua (2020):
RO-Crate RDA maDMP Mapper.
Zenodo
https://github.com/GhaithArf/ro-crate-rda-madmp-mapper
https://doi.org/10.5281/zenodo.3922136

[Arkisto 2022] Arkisto (2022):
Tools: Data Portal & Discovery.
https://arkisto-platform.github.io/tools/portal/

[Atkinson 2017] Malcolm Atkinson, Sandra Gesing, Johan Montagnat, Ian Taylor (2017):
Scientific workflows: Past, present and future.
Future Generation Computer Systems **75**
https://doi.org/10.1016/j.future.2017.05.041

[Atkinson 2019] Rob Atkinson, Nicholas J. Car (2019):
The Profiles Vocabulary.
W3C Working Group Note
https://www.w3.org/TR/2019/NOTE-dx-prof-20191218/

[Ayris 2016] Paul Ayris, Jean-Yves Berthou, Rachel Bruce, Stefanie Lindstaedt, Anna Monreale, Barend Mons, Yasuhiro Murayama, Caj Södergård, Klaus Tochtermann, Ross Wilkinson (2016):
Realising the European Open Science Cloud. First report and recommendations of the Commission High Level Expert Group of the European Open Science Cloud.
Publications Office of the EU
https://doi.org/10.2777/940154

[Azeroual 2022] Otmane Azeroual, Joachim Schöpfel, Janne Pölönen, Anastasija Nikiforova (2022):
Putting FAIR Principles in the Context of Research Information: FAIRness for CRIS and CRIS for FAIRness.
International Conference on Knowledge Discovery and Information Retrieval
https://hdl.handle.net/11366/2243

[Bacall 2019] Finn Bacall, Stian Soiland-Reyes, Marina Soares e Silva (2019):
eScienceLab: RO-Composer.
https://esciencelab.org.uk/projects/ro-composer/
https://github.com/ResearchObject/research-object-composer

[Bacall 2022] Finn Bacall, Alan R. Williams, Stuart Owen, Stian Soiland-Reyes (2022):
Workflow RO-Crate Profile 1.0.
WorkflowHub community
https://w3id.org/workflowhub/workflow-ro-crate/1.0

[Bacall 2022b] Finn Bacall, Martyn Whitwell (2022):
GitHub – ResearchObject/ro-crate-ruby: A Ruby gem for creating, manipulating and reading RO-Crates.
https://github.com/ResearchObject/ro-crate-ruby

[Bahra 2011] Avi Bahra (2011):
Managing work flows with ecFlow.
ECMWF Newsletter **129**
https://doi.org/10.21957/nr843dob

[Bahui 2020] Christophe Bahim, Carlos Casorrán-Amilburu, Makx Dekkers, Edit Herczog, Nicolas Loozen, Konstantinos Repanas, Keith Russell, Shelley Stall (2020):
The FAIR data maturity model: An approach to harmonise FAIR assessments.
Data Science Journal **19**(1)
https://doi.org/10.5334/dsj-2020-041

[Baker 2013] Thomas Baker, Sean Bechhofer, Antoine Isaacc, Alistair Miles, Guus Schreiber, Ed Summers (2013):
Key choices in the design of Simple Knowledge Organization System (SKOS).
Journal of Web Semantics **20**
https://doi.org/10.1016/j.websem.2013.05.001

[Baker 2016] Monya Baker (2016):
1,500 scientists lift the lid on reproducibility.
Nature **533**
https://doi.org/10.1038/533452a

[Baker 2019] Thomas Baker, Eric Prud'hommeaux (2019):
Shape Expressions (ShEx) 2.1 Primer.
http://shex.io/shex-primer/ (accessed 26 May 2022)

[Baker 2020] Dannon Baker, Marius van den Beek, Daniel Blankenberg, Dave Bouvier, John Chilton, Nate Coraor, Frederik Coppens, Ignacio Eguinoa, Simon Gladman, Björn Grüning, Nicholas Keener, Delphine Larivière, Andrew Lonie, Sergei Kosakovsky Pond, Wolfgang Maier, Anton Nekrutenko, James Taylor, Steven Weaver (2020):
No more business as usual: Agile and effective responses to emerging pathogen threats require open data and open analytics.
PLOS Pathogens **16**(8):e1008643
https://doi.org/10.1371/journal.ppat.1008643

[Barker 2019] Michelle Barker, Ross Wilkinson, Andrew Treloar (2019):
The Australian Research Data Commons.
Data Science Journal **18**(1)
https://doi.org/10.5334/dsj-2019-044

[Barker 2022] Michelle Barker, Neil P. Chue Hong, Daniel S. Katz, Anna-Lena Lamprecht, Carlos Martinez-Ortiz, Fotis Psomopoulos, Jennifer Harrow, Leyla Jael Castro, Morane Gruenpeter, Paula Andrea Martinez, Tom Honeyman (2022):
Introducing the FAIR Principles for research software.
Scientific Data **9**:622
https://doi.org/10.1038/s41597-022-01710-x

[Batista 2022] Dominique Batista, Alejandra Gonzalez-Beltran, Susanna-Assunta Sansone, Philippe Rocca-Serra (2022):
Machine actionable metadata models.
Scientific Data **9**:592
https://doi.org/10.1038/s41597-022-01707-6

[Baxter 2012] Robert Baxter, Neil Chue Hong, Dirk Gorissen, James Hetherington, Ilian Todorov (2012):
The Research Software Engineer.
Digital Research 2012, 2012-09-10/–12, Oxford, UK
https://www.research.ed.ac.uk/en/publications/e8416ad7-750f-442f-9b17-d812b9bb414d

[Bayarri 2021a] Genís Bayarri, Robin Long (2021):
Protein MD Setup tutorial using BioExcel Building Blocks (biobb) in CWL.
WorkflowHub. Workflow (CWL).
https://doi.org/10.48546/workflowhub.workflow.29.3

[Bayarri 2021b] Genís Bayarri, Adam Hospital, Douglas Lowe (2021):
Protein MD Setup tutorial using BioExcel Building Blocks (biobb) in Jupyter Notebook.
WorkflowHub. Workflow (Jupyter Notebook).
https://doi.org/10.48546/workflowhub.workflow.120.2

[Bayarri 2022] Genís Bayarri, Adam Hospital (2022):
CWL GMX Automatic Ligand Parameterization tutorial.
Workflow Hub (Common Workflow Language)
https://doi.org/10.48546/workflowhub.workflow.255.1

[Bechhofer 2013] Sean Bechhofer, Iain Buchan, David De Roure, Paolo Missier, John Ainsworth, Jiten Bhagat, Phillip Couch, Don Cruickshank, Mark Delderfield, Ian Dunlop, Matthew Gamble, Danius Michaelides, Stuart Owen, David Newman, Shoaib Sufi, Carole Goble (2013):
Why Linked Data is not enough for scientists.
Future Generation Computer Systems **29**(2) pp. 599–611.
https://doi.org/10.1016/j.future.2011.08.004

[Beg 2021] Marijan Beg, Juliette Taka, Thomas Kluyver, Alexander Konovalov, Min Ragan-Kelley, Nicolas M. Thiery, Hans Fangohr (2021):
Using Jupyter for Reproducible Scientific Workflows.
Computing in Science & Engineering **23**(2) pp 36–46.
https://doi.org/10.1109/MCSE.2021.3052101

[Beier 2024] Sebastian Beier, Timo Mühlhaus, Cyril Pommier, Stuart Owen, Dominik Brilhaus, Heinrich Lukas Weil, Florian Wetzels, Gavin Chait, Daniel Arend, Manuel Feser, Gajendra Doniparthi, Jonathan Bauer, Sveinung Gundersen, Pável Vázquez (2024):
BioHackEU23 report: **Enabling continuous RDM using Annotated Research Contexts with RO-Crate profiles for ISA**.
BioHackrXiv
https://doi.org/10.37044/osf.io/7y2jh

[Belchev 2021] Kostadin Belchev (2021):
KockataEPich/CheckMyCrate: A command line application for validating a RO-Crate object against a JSON profile.
GitHub
https://github.com/KockataEPich/CheckMyCrate

[Belhajjame 2015] Khalid Belhajjame, Jun Zhao, Daniel Garijo, Matthew Gamble, Kristina Hettne, Raul Palma, Eleni Mina, Oscar Corcho, José Manuel Gómez-Pérez, Sean Bechhofer, Graham Klyne, Carole Goble (2015):
Using a suite of ontologies for preserving workflow-centric research objects.
Web Semantics: Science, Services and Agents on the World Wide Web **32** pp. 16–42.
https://doi.org/10.1016/j.websem.2015.01.003

[Belshe 2022] Mike Belshe, Roberto Peon, Martin Thomson (2015):
Hypertext Transfer Protocol Version 2 (HTTP/2).
RFC Editor, RFC 7540
https://doi.org/10.17487/rfc7540

[Beltrán 2023] Daniel Beltrán Mora, Miguel Castrillo, Manuel G. Marciani, Bruno P. Kinoshita, Luiggi Tenorio Ku, Aina Gaya-Àvila, Francesc Roura Adserias, Pierre-Antoine Bretonnière, Oriol Mula Valls, Pablo Goitia, Julian Rodrigo Berlin, Miguel Andrés-Martínez, Kim Serradell, Wilmer Uruchi Ticona, Domingo Manubens-Gil, Larissa Batista Leite, Isabel Andreu-Burillo, Javier Vegas-Regidor, Hui Du, Danila Volpi, Fabian Lienert, Joan Giralt, Joan Lopez, Muhammad Asif, Virginie Guemas, Xavier Abellan Ecija (2023):
Autosubmit v4.0.100.
Zenodo
https://doi.org/10.5281/zenodo.10199020

[Benureau 2017] Fabien C. Y. Benureau, Nicolas P. Rougier (2017):
Re-run, repeat, reproduce, reuse, replicate: Transforming code into scientific contributions.
Frontiers in Neuroinformatics **11**:69.
https://doi.org/10.3389/fninf.2017.00069

[Berman 2007] Helen Berman, Kim Henrick, Haruki Nakamura, John L Markley (2007):
The worldwide Protein Data Bank (wwPDB): Ensuring a single, uniform archive of PDB data.
Nucleic Acids Research **35**(Database issue), D301–D303.
https://doi.org/10.1093/nar/gkl971

[Berners-Lee 1998] Tim Berners-Lee (1998):
Cool URIs don't change.
Style Guide for online hypertext, W3C
https://www.w3.org/Provider/Style/URI

[Berners-Lee 1999] Tim Berners-Lee, Mark Fischetti (1999):
Weaving the Web: The original design and ultimate destiny of the World Wide Web by its inventor.
ISBN 978-0-06-251586-5

[Berners-Lee 2000] Tim Berners-Lee (2000):
Semantic Web on XML.
XML 2000, Washington DC, 2000-12-06.
https://www.w3.org/2000/Talks/1206-xml2k-tbl/slide10-0.html (accessed 24 January 2023)

[Berners-Lee 2005] Tim Berners-Lee, Roy T. Fielding, Larry M. Masinter (2005):
Uniform Resource Identifier (URI): Generic Syntax.
RFC Editor, RFC 3986
https://doi.org/10.17487/rfc3986

[Berners-Lee 2006] Tim Berners-Lee (2006):
Linked Data
Design Issues, W3C
https://www.w3.org/DesignIssues/LinkedData.html

[Bernstein 2016] Abraham Bernstein, James Hendler, Natalya Noy (2016):
A new look at the semantic web.
Communications of the ACM **59**(9)
https://doi.org/10.1145/2890489

[Bietrix 2021] Florence Bietrix, José Maria Carazo, Salvador Capella-Gutierrez, Frederik Coppens, Maria Luisa Chiusano, Romain David, Jose Maria Fernandez, Maddalena Fratelli, Jean-Karim Heriche, Carole Goble, Philip Gribbon, Petr Holub, Robbie Joosten, Simone Leo, Stuart Owen, Helen Parkinson, Roland Pieruschka, Luca Pireddu, Luca Porcu, Michael Raess, Laura Rodriguez-Navas, Andreas Scherer, Stian Soiland-Reyes, Jing Tang (2021):
EOSC-Life methodology framework to enhance reproducibility within EOSC Life.
Zenodo
https://doi.org/10.5281/zenodo.4705078

[BioMoby 2008] The BioMoby Consortium (2008):
Interoperability with Moby 1.0—It's better than sharing your toothbrush!
Briefings in Bioinformatics **9**(3) pp 220–231.
https://doi.org/10.1093/bib/bbn003

[Bishop 2022] Mike Bishop (2022):
HTTP/3
RFC Editor, RFC 9114
https://doi.org/10.17487/rfc9114

[Bisol 2014] Giovanni Destro Bisol, Paolo Anagnostou, Marco Capocasa, Silvia Bencivelli, Andrea Cerroni, Jorge Contreras, Neela Enke, Bernardino Fantini, Pietro Greco, Catherine Heeney, Daniela Luzi, Paolo Manghi, Deborah Mascalzoni, Jennifer C. Molloy, Fabio Parenti, Jelte M Wicherts, Geoffrey Boulton (2014):
Perspectives on Open Science and Scientific Data Sharing: An Interdisciplinary Workshop.
Journal of Anthropological Sciences **92**
https://www.isita-org.com/jass/Contents/2014vol92/Destro/25020017.pdf
https://doi.org/10.4436/JASS.92006

[Bizer 2009] Christian Bizer, Tom Heath, Tim Berners-Lee (2009):
Linked Data - The Story So Far.
International Journal on Semantic Web and Information Systems **5**(3)
https://doi.org/10.4018/jswis.2009081901

[Bizer 2011] Christian Bizer, Tom Heath, Tim Berners-Lee (2011):
Linked data: The story so far.
In *Semantic Services, Interoperability and Web Applications: Emerging Concepts*, Amit Sheth (ed.) ISBN 9781609605933
https://doi.org/10.4018/978-1-60960-593-3.ch008

[Blanchi 2022] Christophe Blanchi, Daan Broeder, Thomas Jejkal, Islam Sharif, Alexander Schlemmer, Dieter van Uytvanck, Peter Wittenburg (2022):
FDO – upload of FDO.
FDO Specification Documents PEN-FDO-Upload-1.1-20221017
FAIR Digital Objects Forum
https://doi.org/10.5281/zenodo.7825549

[Blanchi 2023] Christophe Blanchi, Maggie Hellström, Larry Lannom, Andreas Pfeil, Ulrich Schwardmann, Peter Wittenburg (2022):
Implementation of attributes, types, profiles and registries.
FDO Specification Documents WD-Implementation-of-Attributes-0.4-20230314
FAIR Digital Objects Forum
https://doi.org/10.5281/zenodo.7825572

[Blankenberg 2014] Daniel Blankenberg, Gregory Von Kuster, Emil Bouvier, Dannon Baker, Enis Afgan, Nicholas Stoler, James Taylor, Anton Nekrutenko, the Galaxy Team (2014):
Dissemination of scientific software with Galaxy ToolShed.
Genome Biology **15**:403
https://doi.org/10.1186/gb4161

[Blomquist 2009] Eva Blomqvist, Aldo Gangemi, Valentina Presutti (2009):
Experiments on pattern-based ontology design.
K-CAP '09: Proceedings of the fifth international conference on Knowledge capture
https://doi.org/10.1145/1597735.1597743

[Bonino 2016] Luiz Olavo Bonino Da Silva Santos, Mark D. Wilkinson, Arnold Kuzniar, Rajaram Kaliyaperumal, Mark Thompson, Michel Dumontier, Kees Burger (2016):
FAIR Data points supporting big data interoperability.
Enterprise interoperability in the digitized and networked factory of the future, Martin Zelm, Guy Doumeingts, Joao Pedro Mendonça (eds.).
iSTE Press.
ISBN 978-1-84704-044-2
Preprint: https://www.researchgate.net/publication/309468587_FAIR_Data_Points_Supporting_Big_Data_Interoperability

[Bonino 2019] Luiz Bonino, Peter Wittenburg, Bonnie Carroll, Alex Hardisty, Mark Leggott, Carlo Zwölf (2019):
FAIR digital object framework v1.02.
FDOF technical implementation guideline.
Group of European Data Experts in RDA (GEDE-RDA)
https://github.com/GEDE-RDA-Europe/GEDE/blob/master/FAIR%20Digital%20Objects/FDOF/FAIR%20Digital%20Object%20Framework-v1-02.docx

[Bonino 2020] Luiz Olavo Bonino da Silva Santos, Giancarlo Guizzardi, Tiago Prince Sales (2022):
FAIR Digital Object Framework Documentation.
27 October 2022
https://fairdigitalobjectframework.org/

[Bouyssié 2023] David Bouyssié, Pınar Altıner, Salvador Capella-Gutierrez, José M. Fernández, Yanick Paco Hagemeijer, Peter Horvatovich, Martin Hubálek, Fredrik Levander, Pierluigi Mauri, Magnus Palmblad, Wolfgang Raffelsberger, Laura Rodríguez-Navas, Dario Di Silvestre, Balázs Tibor Kunkli, Julian Uszkoreit, Yves Vandenbrouck, Juan Antonio Vizcaíno, Dirk Winkelhardt, Veit Schwämmle (2023):
WOMBAT-P: Benchmarking Label-Free Proteomics Data Analysis Workflows.
bioRxiv 2023.10.02.560412
https://doi.org/10.1101/2023.10.02.560412

[Brack 2022a] Paul Brack, Peter Crowther, Stian Soiland-Reyes, Stuart Owen, Douglas Lowe, Alan R Williams, Quentin Groom, Mathias Dillen, Frederik Coppens, Björn Grüning, Ignacio Eguinoa, Phil Ewels, Carole Goble (2022):
Ten Simple Rules for making a software tool workflow-ready.
PLOS Computational Biology **18**(3):e1009823
https://doi.org/10.1371/journal.pcbi.1009823
https://s11.no/2022/phd/10-simple-rules-for-workflow-tools/ (Supplement 1)

[Brack 2022b] Paul Brack, Oliver Woolland, Laurence Livermore (2022):
De novo digitisation. (Galaxy workflow)
WorkflowHub
https://doi.org/10.48546/workflowhub.workflow.373.1

[Brand 2015] Amy Brand, Liz Allen, Micah Altman, Marjorie Hlava, Jo Scott (2015):
Beyond authorship: Attribution, contribution, collaboration, and credit.
Learned Publishing **28**(2) pp. 151–155.
https://doi.org/10.1087/20150211

[Bray 2017] Tim Bray (2017):
The JavaScript Object Notation (JSON) Data Interchange Format.
STD 90, RFC 8259
RFC Editor, Internet Engineering Task Force.
https://doi.org/10.17487/rfc8259

[Brenner 2020] Gabriel Brenner (2020):
BrennerG/Ro-Crate_2_ma-DMP: v1.0.0.
https://github.com/BrennerG/Ro-Crate_2_ma-DMP
https://doi.org/10.5281/zenodo.3903463

[Brickley 2014] Dan Brickley, Libby Miller (2014):
FOAF Vocabulary Specification.
http://xmlns.com/foaf/spec/ (accessed 26 May 2022)

[Broeder 2022] Daan Broeder, Peter Wittenburg (2022):
FDO glossary november 2022.
FDO Specification Documents (internal draft)
FAIR Digital Objects Forum
https://drive.google.com/file/d/1KJ9l0p96naKi_2HPJ_MPqPTwS_zlP92G (accessed 2 February 2023)

[Buck 2022] Justin Buck, Deb Agarwal, James Ayliffe, Chris Erdmann, Carole Goble, Ugis Sarkans, Daniel Noesgaard, Uwe Schindler, Shelley Stall, Martin Fenner, Martina Stockhause, Paolo Manghi (2022):
AGU data citation community of practice - Credit for creators of data within collections using the concept of a reliquary.
ESS Open Archive
https://doi.org/10.1002/essoar.10509966.1

[Capadisli 2017] Sarven Capadisli, Amy Guy, eds. (2017):
Linked Data Notifications.
W3C Recommendation 2 May 2017
https://www.w3.org/TR/2017/REC-ldn-20170502/

[Carballo-Garcia 2022] Ana Carballo-Garcia, Juan-José Boté-Vericad (2022):
FAIR Data: History and Present Context.
Central European Journal of Educational Research **4**(2)
https://doi.org/10.37441/cejer/2022/4/2/11379

[Cardoso 2020a] João Cardoso, Diogo Proença, José Borbinha (2020):
Machine-actionable data management plans: A knowledge retrieval approach to automate the assessment of funders' requirements.
ECIR 2020: Advances in Information Retrieval
ISBN 978-3-030-45442-5.
https://doi.org/10.1007/978-3-030-45442-5_15

[Cardoso 2020b] João Cardoso, Leyla Jael Garcia Castro, Fajar Ekaputra, Marie-Christine Jacquemot-Perbal, Tomasz Miksa, José Borbinha (2020):
Towards Semantic Representation of Machine-Actionable Data Management Plans.
PUBLISSO
https://repository.publisso.de/resource/frl:6423289
https://doi.org/10.4126/frl01-006423289

[Carranza-Rojas 2017] Jose Carranza-Rojas, Herve Goeau, Pierre Bonnet, Erick Mata-Montero, Alexis Joly (2017):
Going deeper in the automated identification of Herbarium specimens.
BMC Evolutionary Biology **17**(1)
https://doi.org/10.1186/s12862-017-1014-z

[Carriero 2010] Valentina Anita Carriero, Marilena Daquino, Aldo Gangemi, Andrea Giovanni Nuzzolese, Silvio Peroni, Valentina Presutti, Francesca Tomasi (2020):
The landscape of ontology reuse approaches.
Applications and practices in ontology design, extraction, and reasoning
https://doi.org/10.3233/ssw200033

[Carrothers 2014] Gavin Carothers, Andy Seaborne, David Beckett (2014):
RDF 1.1 N-Triples: A line-based syntax for an RDF graph.
W3C Recommendation 25 February 2014
http://www.w3.org/TR/2014/REC-n-triples-20140225/

[Chard 2014] Kyle Chard, Steven Tuecke, Ian Foster (2014):
Efficient and secure transfer, synchronization, and sharing of big data.
IEEE Cloud Computing **1**(3) pp. 46–55.
https://doi.org/10.1109/MCC.2014.52

[Chard 2016] Kyle Chard, Mike D' Arcy, Ben Heavner, Ian Foster, Carl Kesselman, Ravi Madduri, Alexis Rodriguez, Stian Soiland-Reyes, Carole Goble, Kristi Clark, Eric W. Deutsch, Ivo Dinov, Nathan Price, Arthur Toga (2016):
I'll take that to go: Big data bags and minimal identifiers for exchange of large, complex datasets.
2016 IEEE International Conference on Big Data (*Big Data*), IEEE, pp. 319–328.
ISBN 978-1-4673-9005-7.
https://static.aminer.org/pdf/fa/bigdata2016/BigD418.pdf
https://doi.org/10.1109/BigData.2016.7840618

[Chard 2019] Kyle Chard, Niall Gaffney, Matthew B. Jones, Kacper Kowalik, Bertram Ludascher, Timothy McPhillips, Jarek Nabrzyski, Victoria Stodden, Ian Taylor, Thomas Thelen, Matthew J. Turk, Craig Willis (2019):
Application of BagIt-serialized research object bundles for packaging and re-execution of computational analyses.
15th International Conference on eScience (*eScience 2019*), IEEE, pp. 514–521.
ISBN 978-1-7281-2451-3.
https://zenodo.org/record/3381754
https://doi.org/10.1109/eScience.2019.00068

[Chard 2020] Kyle Chard, Niall Gaffney, Mihael Hategan, Kacper Kowalik, Bertram Ludäscher, Timothy McPhillips, Jarek Nabrzyski, Victoria Stodden, Ian Taylor, Thomas Thelen, Matthew J. Turk, Craig Willis (2020):
Toward enabling reproducibility for data-intensive research using the Whole Tale platform.
Advances in Parallel Computing **36** pp 766–778.
https://doi.org/10.3233/APC200107

[Ciccarese 2013] Paolo Ciccarese, Stian Soiland-Reyes, Khalid Belhajjame, Alasdair JG Gray, Carole Goble, Tim Clark (2013):
PAV ontology: Provenance, authoring and versioning.
Journal of Biomedical Semantics **4**(1):37.
https://doi.org/10.1186/2041-1480-4-37

[Ciccarese 2017] Paolo Ciccarese, Robert Sanderson, Benjamin Young (2017):
Web Annotation Data Model.
W3C Recommendation 23 February 2017.
https://www.w3.org/TR/2017/REC-annotation-model-20170223/

[Claerbout 1992] Jon F. Claerbout, Martin Karrenbach (1992):
Electronic documents give reproducible research a new meaning.
SEG Technical Program Expanded Abstracts 1992, Society of Exploration Geophysicists, pp. 601–604.
http://sep.stanford.edu/oldsep/matt/join/redoc/web/seg92.html
https://doi.org/10.1190/1.1822162

[Clemm 2002] Geoffrey M. Clemm, Jim Amsden, Tim Ellison, Christopher Kaler, Jim Whitehead (2002):
Versioning Extensions to WebDAV (Web Distributed Authoring and Versioning).
RFC Editor, RFC 3253
https://doi.org/10.17487/rfc3253

[CNRI 2023a] CNRI (2023):
DOIP API for HTTP Clients.
Cordra® Software Technical Manual Version 2.5.0
Corporation for National Research Initiatives.
https://www.cordra.org/documentation/api/doip-api-for-http-clients.html (accessed 13 June 2023)

[CNRI 2023b] CNRI (2023):
DOIP and Examples.
Cordra® Software Technical Manual Version 2.5.0
Corporation for National Research Initiatives.
https://www.cordra.org/documentation/api/doip.html (accessed 14 June 2023)

[Cohen 2020] Jeremy Cohen, Daniel S. Katz, Michelle Barker, Neil Chue Hong, Robert Haines, Caroline Jay (2020):
The Four Pillars of Research Software Engineering.
IEEE Software **38**(1)
arXiv:2002.01035
https://doi.org/10.1109/MS.2020.2973362

[Cohen-Boulakia 2017] Sarah Cohen-Boulakia, Khalid Belhajjame, Olivier Collin, Jérôme Chopard, Christine Froidevaux, Alban Gaignard, Konrad Hinsen, Pierre Larmande, Yvan Le Bras, Frédéric Lemoine, Fabien Mareuil, Hervé Ménager, Christophe Pradal, Christophe Blanchet (2017):
Scientific workflows for computational reproducibility in the life sciences: Status, challenges and opportunities.
Future Generation Computer Systems **75** pp. 284–298.
https://hal.archives-ouvertes.fr/hal-01516082
https://doi.org/10.1016/j.future.2017.01.012

[Colonnelli 2021] Iacopo Colonnelli, Barbara Cantalupo, Ivan Merelli, Marco Aldinucci (2021):
StreamFlow: cross-breeding Cloud with HPC.
IEEE Transactions on Emerging Topics in Computing **9**(4)
https://doi.org/10.1109/TETC.2020.3019202

[Colonnelli 2023a] Iacopo Colonnelli (2023):
StreamFlow run of digital pathology tissue/tumor prediction workflow.
Zenodo
https://doi.org/10.5281/zenodo.7911906

[Colonnelli 2023b] Iacopo Colonnelli, Barbara Cantalupo, Marco Aldinucci, Gaetano Saitta, Alberto Mulone (2023):
alpha-unito/streamflow version 0.2.0.dev10
GitHub
https://github.com/alpha-unito/streamflow/releases/tag/0.2.0.dev10
https://identifiers.org/swh:1:rev:b2014add57189900fa5a0a0403b7ae3a384df73b

[Corcho 2021] Oscar Corcho, Esteban González, Daniel Garijo, Raul Palma (2021):
D5.1 RO Model Adapted to EOSC.
RELIANCE deliverable, *Zenodo*
https://doi.org/10.5281/zenodo.4913285

[Corcho 2023] Oscar Corcho, Fajar J. Ekaputra, Ivan Heibi, Clement Jonquet, Andras Micsik, Silvio Peroni, Emanuele Storti (2023):
A maturity model for catalogues of semantic artefacts.
arXiv:2305.06746 [cs.DL]
https://doi.org/10.48550/arXiv.2305.06746

[Cossu 2018] Stefano Cossu, Esmé Cowles, Karen Estlund, Christina Harlow, Tom Johnson, Mark Matienzo, Danny Lamb, Lynette Rayle, Rob Sanderson, Jon Stroop, Andrew Woods (2018):
Portland Common Data Model.
GitHub duraspace/pcdm Wiki (2018-06-15)
https://github.com/duraspace/pcdm/wiki

[Costa 2013] Flavio Costa, Vítor Silva, Daniel de Oliveira, Kary Ocaña, Eduardo Ogasawara, Jonas Dias, Marta Mattoso (2013):
Capturing and querying workflow runtime provenance with PROV: a practical approach.
Proceedings of the Joint EDBT/ICDT 2013 Workshops (*EDBT '13*)
https://doi.org/10.1145/2457317.2457365

[Crosas 2011] Mercè Crosas (2011):
The DataVerse Network: An open-source application for sharing, discovering and preserving data.
D-Lib Magazine **17**(1/2)a
https://doi.org/10.1045/january2011-crosas

[Crosas 2020] Mercè Crosas (2020):
Harvard Data Commons.
European Dataverse Workshop 2020, Tromsø, Norway. ISSN 2387-3086.
https://doi.org/10.7557/5.5422

[Crosswell 2012] Lindsey C Crosswell, Janet M Thornton (2012):
ELIXIR: A distributed infrastructure for European biological data.
Trends in Biotechnology **30**(5) pp. 241–242.
https://doi.org/10.1016/j.tibtech.2012.02.002

[CRS4 2022] CRS4 (2022):
LifeMonitor, a testing and monitoring service for scientific workflows.
https://about.lifemonitor.eu/

[Crusoe 2022] Michael R. Crusoe, Sanne Abeln, Alexandru Iosup, Peter Amstutz, John Chilton, Nebojša Tijanić, Hervé Ménager, Stian Soiland-Reyes, Bogdan Gavrilović, Carole Goble, The CWL Community (2022):
Methods Included: Standardizing Computational Reuse and Portability with the Common Workflow Language.
Communications of the ACM **65**(6)
https://doi.org/10.1145/3486897
https://s11.no/2022/phd/methods-included/ (Supplement 4)

[Cruz 2009] Sérgio Manuel Serra da Cruz; Maria Luiza M. Campos; Marta Mattoso (2009):
Towards a Taxonomy of Provenance in Scientific Workflow Management Systems.
IEEE World Congress on Services (SERVICES) 2009 **I**
https://doi.org/10.1109/SERVICES-I.2009.18

[Cuevas-Vicenttín 2016] Víctor Cuevas-Vicenttín, Bertram Ludäscher, Paolo Missier, Khalid Belhajjame, Fernando Chirigati, Yaxing Wei, Saumen Dey, Parisa Kianmajd, David Koop, Shawn Bowers, Ilkay Altintas, Christopher Jones, Matthew B. Jones, Lauren Walker, Peter Slaughter, Ben Leinfelder, Yang Cao (2016):
ProvONE: A PROV Extension Data Model for Scientific Workflow Provenance.
https://purl.dataone.org/provone-v1-dev
(accessed 2023-11-06)

[CWFR 2021] The CWFR Group, Alex Hardisty (ed.), Peter Wittenburg (ed.) (2021):
Canonical Workflow Frameworks for Research.
Position Paper, version 2 2021-01-06.
OSF
https://osf.io/3rekv/

[da Veiga Leprevost 2017] Felipe da Veiga Leprevost, Björn A Grüning, Saulo Alves Aflitos, Hannes L Röst, Julian Uszkoreit, Harald Barsnes, Marc Vaudel, Pablo Moreno, Laurent Gatto, Jonas Weber, Mingze Bai, Rafael C Jimenez, Timo Sachsenberg, Julianus Pfeuffer, Roberto Vera Alvarez, Johannes Griss, Alexey I Nesvizhskii, Yasset Perez-Riverol (2017):
BioContainers: An open-source and community-driven framework for software standardization.
Bioinformatics **33**(16)
https://doi.org/10.1093/bioinformatics/btx192

[Davidson 2019] Joy Davidson, Claudia Engelhardt, Vanessa Proudman, Lennart Stoy, Angus Whyte (2019):
D3.1 FAIR Policy Landscape Analysis.
FAIRsFAIR project deliverable
https://doi.org/10.5281/zenodo.5537032

[Davidson 2022] Joy Davidson, Angus Whyte, Laurence Horton, Marjan Grootveld (2022):
D3.8 Final report on policy and practice recommendations and support.
FAIRsFAIR project deliverable
https://doi.org/10.5281/zenodo.6699333

[DCMI 2020] DCMI Usage Board (2020):
DCMI Metadata Terms.
DCMI Recommendation
https://www.dublincore.org/specifications/dublin-core/dcmi-terms/2020-01-20/

[De Geest 2022] Paul De Geest, Frederik Coppens, Stian Soiland-Reyes, Ignacio Eguinoa, Simone Leo (2022):
Enhancing RDM in Galaxy by integrating RO-Crate.
1st International Conference on FAIR Digital Objects (FDO 2022) (poster)
Research Ideas and Outcomes **8**:e95164
https://doi.org/10.3897/rio.8.e95164
https://s11.no/2022/phd/galaxy-ro-crate/ (Supplement 2)

[De Geest 2023a] Paul De Geest, Bert Droesbeke, Ignacio Eguinoa, Alban Gaignard, Sebastiaan Huber, Bruno Kinoshita, Simone Leo, Luca Pireddu, Laura Rodríguez-Navas, Raül Sirvent, Stian Soiland-Reyes (2023):
GitHub – ResearchObject/ro-crate-py: Python library for RO-Crate, version 0.9.0.
https://github.com/researchobject/ro-crate-py
https://doi.org/10.5281/zenodo.10017862

[De Geest 2023b] Paul De Geest (2023):
Run of an example Galaxy collection workflow.
Zenodo
https://doi.org/10.5281/zenodo.7785861

[De Giovanni 2016] Renato De Giovanni, Alan R. Williams, Vera Hernández Ernst, Robert Kulawik, Francisco Quevedo Fernandez, Alex R. Hardisty (2016):
ENM Components: a new set of web service-based workflow components for ecological niche modelling.
Ecography **39**(4) **pp** 376–383.
https://doi.org/10.1111/ecog.01552

[De Roure 2010] David De Roure, Carole Goble (2010):
Anchors in shifting sand: the primacy of method in the web of data.
Proceedings of the WebSci10: Extending the Frontiers of Society On-Line, at Web Science Conference 2010 Raleigh, NC: US 2010-04-26/–27
https://web.archive.org/web/20140828142306/http://journal.webscience.org/325/
http://eprints.soton.ac.uk/id/eprint/270817

[De Smedt 2020] Koenraad De Smedt, Dimitris Koureas, Peter Wittenburg (2020):
FAIR Digital Objects for Science: From Data Pieces to Actionable Knowledge Units.
Publications **8**(2):21
https://doi.org/10.3390/publications8020021

[Del Rio 2022] Mauro Del Rio, Luca Lianas, Oskar Aspegren, Giovanni Busonera, Francesco Versaci, Renata Zelic, Per H. Vincent, Simone Leo, Andreas Pettersson, Olof Akre, Luca Pireddu (2022):
AI Support for Accelerating Histopathological Slide Examinations of Prostate Cancer in Clinical Studies.
Image Analysis and Processing. ICIAP 2022 Workshops. ICIAP 2022.
Lecture Notes in Computer Science **13373**
https://doi.org/10.1007/978-3-031-13321-3_48

[Delgado 2016] José Carlos Martins Delgado (2016):
An Interoperability Framework and Distributed Platform for Fast Data Applications.
Data Science and Big Data Computing
https://doi.org/10.1007/978-3-319-31861-5_1

[Desai 2016] Tanvi Desai, Felix Ritchie, Richard Welpton (2016):
Five Safes: designing data access for research.
Economics Working Paper Series **1601**
https://econpapers.repec.org/RePEc:uwe:wpaper:20161601

[Devaraju 2021] Anusuriya Devaraju, Mustapha Mokrane, Linas Cepinskas, Robert Huber, Patricia Herterich, Jerry de Vries, Vesa Akerman, Hervé L'Hours, Joy Davidson, Michael Diepenbroek (2021):
From conceptualization to implementation: FAIR assessment of research data objects.
Data Science Journal **20**
https://doi.org/10.5334/dsj-2021-004

[Dillen 2019a] Mathias Dillen, Quentin Groom, Donat Agosti, Lars Nielsen (2019):
Zenodo, an archive and publishing repository: A tale of two herbarium specimen pilot projects.
Biodiversity Information Science and Standards **3**:e37080
https://doi.org/10.3897/biss.3.37080

[Dillen 2019b] Mathias Dillen, Quentin Groom, Simon Chagnoux, Anton Güntsch, Alex Hardisty, Elspeth Haston, Laurence Livermore, Veljo Runnel, Leif Schulman, Luc Willemse, Zhengzhe Wu, Sarah Phillips (2019):
A benchmark dataset of herbarium specimen imagfes with label data.
Biodiversity Data Journal **7**:e31817.
https://doi.org/10.3897/BDJ.7.e31817

[Di Tommaso 2017] Paolo Di Tommaso, Maria Chatzou, Evan W Floden, Pablo Prieto Barja, Emilio Palumbo, Cedric Notredame (2017):
Nextflow enables reproducible computational workflows.
Nature Biotechnology **35**(4)
https://doi.org/10.1038/nbt.3820

[DOI 2019] DOI (2019):
DOI Handbook - Resolution.
DOI Handbook
DOI Foundation
https://doi.org/10.1000/182
https://www.doi.org/doi_handbook/3_Resolution.html

[DONA 2018] DONA Foundation (2018):
Digital Object Interface Protocol specification, version 2.0.
DONA Foundation
https://hdl.handle.net/0.DOIP/DOIPV2.0

[DONA 2021] DONA Foundation (2021):
Digital Object Architecture.
https://www.dona.net/node/88 (accessed 2021-08-10)

[Drummond 2006] Nick Drummond, Rob Shearer (2016):
The Open World Assumption – or Sometimes it is nice to know what we don't know.
eSI Workshop: *The Closed World of Databases Meets the Open World of the Semantic Web*
e-Science Institute, Edinburgh, 2006-10-12
https://www.cs.man.ac.uk/~drummond/presentations/OWA.pdf

[Dürst 2005] Martin J. Dürst, Michel Suignard (2005):
Internationalized resource identifiers (IRIs).
RFC 3987, *Internet Requests for Comments*, RFC Editor
https://doi.org/10.17487/rfc3987

[Duarte 2023] Javier Duarte, Haoyang Li, Avik Roy, Ruike Zhu, E A Huerta, Daniel Diaz, Philip Harris, Raghav Kansal, Daniel S Katz, Ishaan H Kavoori, Volodymyr V Kindratenko, Farouk Mokhtar, Mark S Neubauer, Sang Eon Park, Melissa Quinnan, Roger Rusack, Zhizhen Zhao (2023):
FAIR AI models in high energy physics.
Machine Learning: Science and Technology **4**:4
https://doi.org/10.1088/2632-2153/ad12e3

[Dusseault 2007] Lisa M. Dusseault (2007):
HTTP Extensions for Web Distributed Authoring and Versioning (WebDAV).
RFC Editor, RFC 4918.
https://doi.org/10.17487/rfc4918

[Eguinoa 2020] Ignacio Eguinoa, Stian Soiland-Reyes, Bert Droesbeke, Michael R. Crusoe (2020):
GitHub workflowhub-eu/galaxy2cwl: Standalone version tool to get cwl descriptions (initially an abstract cwl interface) of galaxy workflows and Galaxy workflows executions.
https://github.com/workflowhub-eu/galaxy2cwl

[Eguinoa 2023] Ignacio Eguinoa, Marek Suchánek, Vojtěch Knaisl, Jan Slifka, Paul De Geest, David López, Björn Grüning, Simone Leo, Stian Soiland-Reyes (2023):
BioHackEU22 Report: Enhancing Research Data Management in Galaxy and Data Stewardship Wizard by utilising RO-Crates.
BioHackrXiv
https://doi.org/10.37044/osf.io/24jst
https://s11.no/2023/phd/enhancing-rdm-galaxy-dsw/ (Supplement 18)

[Ejarque 2023] Jorge Ejarque, Francesc Lordan, Rosa Maria Badia, Raul Sirvent, Daniele Lezzi, Fernando Vazquez, Cristian Tatu, Gabriel Puigdemunt, Nihad Mammadli, Javier Conejero (2023):
COMPSs. bsc-wdc/compss. Version 3.2
Zenodo
https://doi.org/10.5281/zenodo.7975340

[Ekuan 2023] Martin Ekuan, Jason Bouska, Mick Alberts, Tim Sherer, Udi Dahan, Mike Kistler, Navarro Ferreira, Theano Petersen, Rodrigo Leite, Donovan Dennis, Alex Buck, Alvaro Enrique Ruano, Dhanas hri Kshirsagar, Asbjørn Ulsberg, David Coulter, Veronica Wasson, Nick Schonning, Marc Wilson, James Watkin, Alexandre Cruz, Christopher Bennage, Luis Gizirian, Francis Cheung (2023):
Web API design best practices.
Azure Architecture Center
https://learn.microsoft.com/en-us/azure/architecture/best-practices/api-design (accessed 24 January 2023)

[Ellerm 2023] Augustus Ellerm, Mark Gahegan, Benjamin Adams (2023):
LivePublication: The Science Workflow Creates and Updates the Publication.
IEEE 19th International Conference on e-Science (e-Science 2023)
https://www.researchgate.net/publication/364107266_Enabling_LivePublication
https://doi.org/10.1109/e-Science58273.2023.10254857

[Ellis 2007] Brian Ellis, Jeffrey Stylos, Brad Myers (2007):
The Factory Pattern in API Design: A Usability Evaluation.
29th International Conference on Software Engineering (ICSE'07)
https://doi.org/10.1109/ICSE.2007.85

[EMBL-EBI 2019] EMBL-EBI Microbiome Informatics Team (2019):
FTP index of /pub/databases/metagenomics/umgs_analyses/.
http://ftp.ebi.ac.uk/pub/databases/metagenomics/umgs_analyses/

[EMBL-EBI 2020] EMBL-EBI Microbiome Informatics Team (2020):
GitHub – Finn-Lab/MGS-gut: Analysing Metagenomic Species (MGS).
https://github.com/Finn-Lab/MGS-gut

[EU 2016] European Commission, Directorate-General for Research & Innovation (2016):
Guidelines on FAIR Data Management in Horizon 2020. Version 3.0
H2020 Programme
https://ec.europa.eu/research/participants/data/ref/h2020/grants_manual/hi/oa_pilot/h2020-hi-oa-data-mgt_en.pdf

[Ewels 2020] Philip A Ewels, Alexander Peltzer, Sven Fillinger, Harshil Patel, Johannes Alneberg, Andreas Wilm, Maxime Ulysse Garcia, Paolo Di Tommaso, Sven Nahnsen (2020):
The nf-core framework for community-curated bioinformatics pipelines.
Nature Biotechnology **38**(3), 276–278.
https://doi.org/10.1038/s41587-020-0439-x

[FAIR Maturity 2020] FAIR Data Maturity Model Working Group (2020):
FAIR data maturity model: Specification and guidelines.
Research Data Alliance
https://doi.org/10.15497/rda00050

[Falk 2010] Raphael Falk (2010):
What is a gene?—Revisited
Studies in History and Philosophy of Biological and Biomedical Sciences **41**(4)
https://doi.org/10.1016/j.shpsc.2010.10.014

[Farnel 2014] Sharon Farnel, Ali Shiri (2014):
Metadata for research data: Current practices and trends.
2014 Proceedings of the International Conference on Dublin Core and Metadata Applications
ISSN 1939-1366.
https://dcpapers.dublincore.org/pubs/article/view/3714

[FDO] FAIR Digital Objects Forum
https://fairdo.org/ (accessed 26 May 2022)

[FDO Specs] FDO (2022):
FDO Specification Documents - November 2022 *FAIR Digital Objects Forum*
https://hdl.handle.net/20.500.14132/fdo-spec-docs
https://fairdo.org/specifications/ (accessed 2 February 2023)

[Feng 2007] Hanhua Feng, Vishal Misra, Dan Rubenstein (2007):
PBS: a unified priority-based scheduler.
Proceedings of the 2007 ACM SIGMETRICS international conference on Measurement and modeling of computer systems (SIGMETRICS '07)
https://doi.org/10.1145/1254882.1254906

[Fenner 2019] Martin Fenner, Amir Aryani (2019):
Introducing the PID graph.
DataCite Blog
https://doi.org/10.5438/jwvf-8a66

[Fensel 2011] Dieter Fensel, Federico Michele Facca, Elena Simperl, Ioan Toma (2011):
Semantic Web Services
https://doi.org/10.1007/978-3-642-19193-0

[Fernández 2023a] José María Fernández, Laura Rodríguez-Navas, Adrián Muñoz-Cívico, Paula Iborra, Daniel Lea (2023):
inab/WfExS-backend. Version 0.10.1
Zenodo
https://doi.org/10.5281/zenodo.10068956

[Fernández 2023b] José María Fernández González (2023):
RO-Crate from staged WfExS working directory 047b6dfc-3547-4e09-92f8-df7143038ff4 (overbridging templon).
Zenodo
https://doi.org/10.5281/zenodo.10091550

[Ferreira da Silva 2023] Rafael Ferreira da Silva, Rosa M. Badia, Venkat Bala, Debbie Bard, Peer-Timo Bremer, Ian Buckley, Silvina Caino-Lores, Kyle Chard, Carole Goble, Shantenu Jha, Daniel S. Katz, Daniel Laney, Manish Parashar, Frederic Suter, Nick Tyler, Thomas Uram, Ilkay Altintas, Stefan Andersson, William Arndt, Juan Aznar, Jonathan Bader, Bartosz Balis, Chris Blanton, Kelly Rosa Braghetto, Aharon Brodutch, Paul Brunk, Henri Casanova, Alba Cervera Lierta, Justin Chigu, Taina Coleman, Nick Collier, Iacopo Colonnelli, Frederik Coppens, Michael Crusoe, Will Cunningham, Bruno de Paula Kinoshita, Paolo Di Tommaso, Charles Doutriaux, Matthew Downton, Wael Elwasif, Bjoern Enders, Chris Erdmann, Thomas Fahringer, Ludmilla Figueiredo, Rosa Filgueira, Martin Foltin, Anne Fouilloux, Luiz Gadelha, Andy Gallo, Artur Garcia Saez, Daniel Garijo, Roman Gerlach, Ryan Grant, Samuel Grayson, Patricia Grubel, Johan Gustafsson, Valerie Hayot-Sasson, Oscar Hernandez, Marcus Hilbrich, AnnMary Justine, Ian Laflotte, Fabian Lehmann, Andre Luckow, Jakob Luettgau, Ketan Maheshwari, Motohiko Matsuda, Doriana Medic, Pete Mendygral, Marek Michalewicz, Jorji Nonaka, Maciej Pawlik, Loic Pottier, Line Pouchard, Mathias Putz, Santosh Kumar Radha, Lavanya Ramakrishnan, Sashko Ristov, Paul Romano, Daniel Rosendo, Martin Ruefenacht, Katarzyna Rycerz, Nishant Saurabh, Volodymyr Savchenko, Martin Schulz, Christine Simpson, Raul Sirvent, Tyler Skluzacek, Stian Soiland-Reyes, Renan Souza, Sreenivas Rangan Sukumar, Ziheng Sun, Alan Sussman, Douglas Thain, Mikhail Titov, Benjamin Tovar, Aalap Tripathy, Matteo Turilli, Bartosz Tuznik, Hubertus van Dam, Aurelio Vivas , Logan Ward, Patrick Widener, Sean Wilkinson, Justyna Zawalska, Mahnoor Zulfiqar (2023):
Workflows Community Summit 2022: A Roadmap Revolution.
Technical Report, ORNL/TM-2023/2885
arXiv:2304.00019 [cs.DC]
https://doi.org/10.48550/arXiv.2304.00019

[Ferreira da Silva 2021] Rafael Ferreira da Silva, Henri Casanova, Kyle Chard, Ilkay Altintas, Rosa M Badia, Bartosz Balis, Tainã Coleman, Frederik Coppens, Frank Di Natale, Bjoern Enders, Thomas Fahringer, Rosa Filgueira, Grigori Fursin, Daniel Garijo, Carole Goble, Dorran Howell, Shantenu Jha, Daniel S. Katz, Daniel Laney, Ulf Leser, Maciej Malawski, Kshitij Mehta, Loïc Pottier, Jonathan Ozik, J. Luc Peterson, Lavanya Ramakrishnan, Stian Soiland-Reyes, Douglas Thain, Matthew Wolf (2021):
A Community Roadmap for Scientific Workflows Research and Development.
2021 IEEE Workshop on Workflows in Support of Large-Scale Science (WORKS)
arXiv:2110.02168
https://doi.org/10.1109/WORKS54523.2021.00016

[Fielding 1999] Roy T. Fielding, Jim Gettys, Jeffrey Mogul, Henrik Frystyk Nielsen, Larry M Masinter, Paul J. Leach, Tim Berners-Lee (1999):
Hypertext Transfer Protocol – HTTP/1.1.
RFC Editor, RFC 2616
https://doi.org/10.17487/rfc2616

[Fielding 2000] Roy Thomas Fielding (2000):
Architectural styles and the design of network-based software architectures
Doctoral Thesis, *University of California*, Irvine.
https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm (accessed 28 June 2022)

[Fielding 2014a] Roy T. Fielding, Julian Reschke (2014):
Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing
RFC Editor, RFC 7230
https://doi.org/10.17487/rfc7230

[Fielding 2014b] Roy T. Fielding, Julian Reschke (2014):
Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content
RFC Editor, RFC 7231
https://doi.org/10.17487/rfc7231

[Fielding 2017] Roy T. Fielding, Richard N. Taylor, Justin R. Erenkrantz, Michael M. Gorlick, Jim Whitehead, Rohit Khare, Peyman Oreizy (2017):
Reflections on the REST architectural style and "principled design of the modern web architecture" (impact paper award).
Proceedings of the 2017 11th joint meeting on foundations of software engineering - ESEC/FSE 2017, New York, New York, USA.
https://doi.org/10.1145/3106237.3121282

[Fielding 2022] Roy T. Fielding, Mark Nottingham, Julian Reschke (2022):
HTTP Semantics.
RFC Editor, RFC 9110
https://doi.org/10.17487/rfc9110

[Fillbrunn 2017] Alexander Fillbrunn, Christian Dietz, Julianus Pfeuffer, René Rahn, Gregory A. Landrum, Michael R. Berthold (2017):
KNIME for reproducible cross-domain analysis of life science data.
Journal of Biotechnology **261**
https://doi.org/10.1016/j.jbiotec.2017.07.028

[Fouilloux 2023] Anne Fouilloux, Elisa Trasatti, Federica Foglini, Alejandro Coca-Castro, Jean Iaquinta (2023):
FAIR Research Objects for realizing Open Science with RELIANCE EOSC project.
Research Ideas and Outcomes **9**:e108765
https://doi.org/10.3897/rio.9.e108765

[Freire 2008] Juliana Freire, David Koop, Emanuele Santos, Cl Silva (2008):
Provenance for Computational Tasks: A Survey.
Computing in Science & Engineering **10**(3)
https://doi.org/10.1109/MCSE.2008.79

[Galaxy 2022] The Galaxy Community (E. Afgan, A. Nekrutenko, B. A. Grüning, D. Blankenberg, J. Goecks, M. C. Schatz, A. E. Ostrovsky, A. Mahmoud, A. J. Lonie, A. Syme, A. Fouilloux, A. Bretaudeau, A. Nekrutenko, A. Kumar, A. C. Eschenlauer, A. D. DeSanto, A. Guerler, B. Serrano-Solano, B. Batut, B. A. Grüning, B. W. Langhorst, B. Carr, B. A. Raubenolt, C. J. Hyde, C. J. Bromhead, C. B. Barnett, C. Royaux, C. Gallardo, D. Blankenberg, D. J. Fornika, D. Baker, D. Bouvier, D. Clements, D. A. de Lima Morais, D. L. Tabernero, D. Lariviere, E. Nasr, E. Afgan, F. Zambelli, F. Heyl, F. Psomopoulos, F. Coppens, G. R. Price, G. Cuccuru, G. L. Corguillé, G. Von Kuster, G. G. Akbulut, H. Rasche, H. Hans-Rudolf, I. Eguinoa, I. Makunin, I. J. Ranawaka, J. P. Taylor, J. Joshi, J. Hillman-Jackson, J. Goecks, J. M. Chilton, K. Kamali, K. Suderman, K. Poterlowicz, L. B. Yvan, L. Lopez-Delisle, L. Sargent, M. E. Bassetti, M. A. Tangaro, M. van den Beek, M. Čech, M. Bernt, M. Fahrner, M. Tekman, M. C. Föll, M. C. Schatz, M. R. Crusoe, M. Roncoroni, N. Kucher, N. Coraor, N. Stoler, N. Rhodes, N. Soranzo, N. Pinter, N. A. Goonasekera, P. A. Moreno, P. Videm, P. Melanie, P. Mandreoli, P. D. Jagtap, Q. Gu, R. J. M. Weber, R. Lazarus, R. H. P. Vorderman, S. Hiltemann, S. Golitsynskiy, S. Garg, S. A. Bray, S. L. Gladman, S. Leo, S. P. Mehta, T. J. Griffin, V. Jalili, V. Yves, V. Wen, V. K. Nagampalli, W. A. Bacon, W. de Koning, W. Maier, P. J. Briggs) (2022):
The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2022 update.
Nucleic Acids Research **50**
https://doi.org/10.1093/nar/gkac247

[Gamma 1995] Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides (1994):
Design Patterns: Elements of Reusable Object-Oriented Software.
Addison Wesley
ISBN 978-0201633610.

[Garcia 2020a] Leyla Garcia, Bérénice Batut, Melissa L. Burke, Mateusz Kuzak, Fotis Psomopoulos, Ricardo Arcila, Teresa K. Attwood, Niall Beard, Denise Carvalho-Silva, Alexandros C. Dimopoulos, Victoria Dominguez del Angel, Michel Dumontier, Kim T. Gurwitz, Roland Krause, Peter McQuilton, Loredana Le Pera, Sarah L. Morgan, Päivi Rauste, Allegra Via, Pascal Kahlem, Gabriella Rustici, Celia W. G. van Gelder, Patricia M. Palagi (2020):
Ten simple rules for making training materials FAIR.
PLOS Computational Biology **16**(5):e1007854
https://doi.org/10.1371/journal.pcbi.1007854

[Garcia 2020b] Leyla Garcia, Erick Antezana, Alexander Garcia, Evan Bolton, Rafael Jimenez, Pjotr Prins, Juan M. Banda, Toshiaki Katayama (2020):
Ten simple rules to run a successful BioHackathon.
PLOS Computational Biology **16**(5):e1007808.
https://doi.org/10.1371/journal.pcbi.1007808

[Garcia-Silva 2019] Andres Garcia-Silva, Jose Manuel Gomez-Perez, Raul Palma, Marcin Krystek, Simone Mantovani, Federica Foglini, Valentina Grande, Francesco De Leo, Stefano Salvi, Elisa Trasatti, Vito Romaniello, Mirko Albani, Cristiano Silvagni, Rosemarie Leone, Fulvio Marelli, Sergio Albani, Michele Lazzarini, Hazel J. Napier, Helen M. Glaves, Timothy Aldridge, Charles Meertens, Fran Boler, Henry W. Loescher, Christine Laney, Melissa A. Genazzio, Daniel Crawl, Ilkay Altintas (2019):
Enabling FAIR research in Earth science through research objects.
Future Generation Computer Systems **98**
arXiv:1809.10617
https://doi.org/10.1016/j.future.2019.03.046

[Garijo 2011] Daniel Garijo, Yolanda Gil (2011):
A New Approach for Publishing Workflows.
Proceedings of the 6th Workshop on Workflows in Support of Large-Scale Science - WORKS '11.
https://doi.org/10.1145/2110497.2110504

[Garijo 2012] Daniel Garijo, Yolanda Gil (2012):
Augmenting PROV with Plans in P-PLAN: Scientific Processes as Linked Data.
Proceedings of the Second International Workshop on Linked Science 2012 - Tackling Big Data. (LISC 2021)
CEUR Workshop Proceedings **951**
https://ceur-ws.org/Vol-951/paper6.pdf

[Garijo 2014a] Daniel Garijo, Pinar Alper, Khalid Belhajjame, Oscar Corcho, Yolanda Gil, Carole Goble (2014):
Common Motifs in Scientific Workflows: An Empirical Analysis.
Future Generation Computer Systems **36** pp 338–351.
https://doi.org/10.1016/j.future.2013.09.018

[Garijo 2014b] Daniel Garijo, Yolanda Gil, Oscar Corcho (2014):
Towards Workflow Ecosystems through Semantic and Standard Representations.
9th Workshop on Workflows in Support of Large-Scale Science (*WORKS 2014*)
https://doi.org/10.1109/works.2014.13

[Gauthier 2019] Jeff Gauthier, Antony T Vincent, Steve J Charette, Nicolas Derome (2019):
A brief history of bioinformatics.
Briefings in Bioinformatics **20**(6)
https://doi.org/10.1093/bib/bby063

[GBIF 2021] GBIF Secretariat. (2021):
GBIF Science Review 2020.
https://doi.org/10.35035/bezp-jj23

[Gewirtz 1996] Paul Gewirtz (1996):
On I Know It When I See It.
Yale Law Journal **105**(4)
https://heinonline.org/HOL/P?h=hein.journals/ylr105&i=1057

[Gil 2011] Yolanda Gil, Varun Ratnakar, Jihie Kim, Pedro Gonzalez-Calero, Paul Groth, Joshua Moody, Ewa Deelman (2011):
Wings: Intelligent Workflow-Based Design of Computational Experiments.
IEEE Intelligent Systems **26**(1)
https://doi.org/10.1109/MIS.2010.9

[Giles 2023] Thomas Giles, Stian Soiland-Reyes, Jonathan Couldridge, Stuart Wheater, Blaise Thomson, Jillian Beggs, Suzy Gallier, Sam Cox, Daniel Lea, Justin Biddle, Rima Doal, Naaman Tammuz, Becca Wilson, Christian Cole, Elizabeth Sapey, Simon Thompson, Professor Emily Jefferson, Phillip Quinlan, Carole Goble (2023):
TRE-FX: Delivering a federated network of trusted research environments to enable safe data analytics.
Zenodo / DARE UK
https://doi.org/10.5281/zenodo.10055354

[GitHub 2021] GitHub (2021):
Managing large files – GitHub Docs.
https://docs.github.com/en/repositories/working-with-files/managing-large-files

[Goble 2008] Carole Goble, Robert Stevens (2008):
State of the nation in data integration for bioinformatics.
Journal of Biomedical Informatics **41**(5)
https://doi.org/10.1016/j.jbi.2008.01.008

[Goble 2010] Carole A Goble, Jiten Bhagat, Sergejs Aleksejevs, Don Cruickshank, Danius Michaelides, David Newman, Mark Borkum, Sean Bechhofer, Marco Roos, Peter Li, David De Roure (2010):
myExperiment: A repository and social network for the sharing of bioinformatics workflows.
Nucleic Acids Research **38**(Web Server issue) W677–W682.
https://doi.org/10.1093/nar/gkq429

[Goble 2016] Carole Goble (2016):
What Is Reproducibility? The R* Brouhaha.
SciRepro Workshop, TPDL, Hannover, Germany, 2016.
http://repscience2016.research-infrastructures.eu/img/CaroleGoble-ReproScience2016v2.pdf

[Goble 2018] Carole Goble, Stian Soiland-Reyes, Sean Bechhofer (2018):
Research Object Community Update.
Workshop on Research Objects (RO 2018), 29 Oct 2018, Amsterdam, Netherlands.
https://doi.org/10.5281/zenodo.1313066

[Goble 2020] Carole Goble, Sarah Cohen-Boulakia, Stian Soiland-Reyes, Daniel Garijo, Yolanda Gil, Michael R. Crusoe, Kristian Peters, Daniel Schober (2020):
FAIR Computational Workflows.
Data Intelligence **2**(1–2) pp 108–121.
https://doi.org/10.1162/dint_a_00033

[Goble 2021] Carole Goble, Stian Soiland-Reyes, Finn Bacall, Stuart Owen, Alan Williams, Ignacio Eguinoa, Bert Droesbeke, Simone Leo, Luca Pireddu, Laura Rodríguez-Navas, José Mª Fernández, Salvador Capella-Gutierrez, Hervé Ménager, Björn Grüning, Beatriz Serrano-Solano, Philip Ewels, Frederik Coppens (2021):
Implementing FAIR Digital Objects in the EOSC-Life Workflow Collaboratory.
Zenodo
https://doi.org/10.5281/zenodo.4605654
https://s11.no/2021/phd/workflow-collaboratory/ (Supplement 3)

[Goble 2022] Carole Goble, Stian Soiland-Reyes, Nick Juty (2022):
FAIR-IMPACT: T3.2.1 PIDs in data production workflows.
FAIR-IMPACT, meeting on T3.2 Integration of PID practices into FAIR data management, 2022-10-06.
Zenodo
https://doi.org/10.5281/zenodo.7157647

[González 2022] Esteban González, Alejandro Benítez, Daniel Garijo (2022):
FAIROs: Towards FAIR Assessment in Research Objects.
TPDL 2022: Linking Theory and Practice of Digital Libraries
https://dgarijo.com/papers/TPDL2022_gonzalez.pdf
https://doi.org/10.1007/978-3-031-16802-4_6

[Gray 2017] Alasdair Gray, Carole Goble, Rafael Jimenez, Bioschemas Community (2017):
Bioschemas: From Potato Salad to Protein Annotation.
Proceedings of the ISWC 2017 posters & demonstrations and industry tracks co-located with 16th international semantic web conference (ISWC 2017), Vienna, Austria.
CEUR Workshop Proceedings **1963**
https://iswc2017.semanticweb.org/paper-579/
https://ceur-ws.org/Vol-1963/paper579.pdf

[Gregorio 2007] Joe Gregorio, Bill de hÓra (2007):
The Atom Publishing Protocol.
RFC Editor, RFC 5023
https://doi.org/10.17487/rfc5023

[Gregorio 2012] Joe Gregorio, Roy T. Fielding, Marc Hadley, Mark Nottingham, David Orchard (2012):
URI Template.
RFC Editor, RFC 6570
https://doi.org/10.17487/rfc6570

[Groom 2020] Quentin Groom, Anton Güntsch, Pieter Huybrechts, Nicole Kearney, Siobhan Leachman, Nicky Nicolson, Roderic D M Page, David P Shorthouse, Anne E Thessen, Elspeth Haston (2020):
People are essential to linking biodiversity data.
Database **2020**
https://doi.org/10.1093/database/baaa072

[Grossman 2016] Robert L Grossman, Allison Heath, Mark Murphy, Maria Patterson, Walt Wells (2016):
A case for data commons: Toward data science as a service.
Computing in Science & Engineering **18**(5)
https://doi.org/10.1109/MCSE.2016.92

[Groth 2014] Paul Groth, Antonis Loizou, Alasdair J. G. Gray, Carole Goble, Lee Harland, Steve Pettifer (2014):
API-centric Linked Data integration: The Open PHACTS Discovery Platform case study.
Journal of Web Semantics **29**
https://doi.org/10.1016/j.websem.2014.03.003

[Grüning 2018a] Björn Grüning, Ryan Dale, Andreas Sjödin, Brad A. Chapman, Jillian Rowe, Christopher H. Tomkins-Tinch, Renan Valieris, the Bioconda Team, Johannes Köster (2018):
Bioconda: Sustainable and Comprehensive Software Distribution for the Life Sciences.
Nature Methods **15** pp 475–476.
https://doi.org/10.1038/s41592-018-0046-7

[Grüning 2018b] Björn Grüning, John Chilton, Johannes Köster, Ryan Dale, Nicola Soranzo, Marius van den Beek, Jeremy Goecks, Rolf Backofen, Anton Nekrutenko, James Taylor (2018):
Practical Computational Reproducibility in the Life Sciences. *Cell Systems* **6**(6)
https://doi.org/10.1016/j.cels.2018.03.014

[Gruenpeter 2021] Morane Gruenpeter, Daniel S. Katz, Anna-Lena Lamprecht, Tom Honeyman, Daniel Garijo, Alexander Struck, Anna Niehues, Paula Andrea Martinez, Leyla Jael Castro, Tovo Rabemanantsoa, Neil P. Chue Hong, Carlos Martinez-Ortiz, Laurents Sesink, Matthias Liffers, Anne Claire Fouilloux, Chris Erdmann, Silvio Peroni, Paula Martinez Lavanchy, Ilian Todorov, Manodeep Sinha (2021):
Defining Research Software: A Controversial Discussion.
FAIR for Research Software (FAIR4RS) / Zenodo
https://doi.org/10.5281/zenodo.5504016

[Guha 2014] Ramanathan Guha, Dan Brickley (2014):
RDF Schema 1.1.
W3C Recommendation
http://www.w3.org/TR/rdf-schema/

[Guha 2015] Ramanathan V Guha, Dan Brickley, Steve Macbeth (2015):
Schema.org: Evolution of Structured Data on the Web: Big data makes common schemas even more necessary.
Queue **13**(9) pp. 10–37.
https://doi.org/10.1145/2857274.2857276

[Guha 2016] Ramanathan V Guha, Dan Brickley, Steve Macbeth (2016):
Schema.org: evolution of structured data on the web.
Communications of the ACM **59**(2)
https://doi.org/10.1145/2844544

[Gurevich 1995] Yuri Gurevich (1995):
Evolving Algebras 1993: Lipari Guide.
Specification and Validation Methods
arXiv:1808.06255
ISBN 978-0198538547

[Handle] CNRI (2022):
Handle.Net Software.
https://www.handle.net/download_hnr.html (accessed 24 January 2023)

[Hardisty 2016] Alex R. Hardisty, Finn Bacall, Niall Beard, Maria-Paula Balcázar-Vargas, Bachir Balech, Zoltán Barcza, Sarah J. Bourlat, Renato De Giovanni, Yde de Jong, Francesca De Leo, Laura Dobor, Giacinto Donvito, Donal Fellows, Antonio Fernandez Guerra, Nuno Ferreira, Yuliya Fetyukova, Bruno Fosso, Jonathan Giddy, Carole Goble, Anton Güntsch, Robert Haines, Vera Hernández Ernst, Hannes Hettling, Dóra Hidy, Ferenc Horváth, Dóra Ittzés, Péter Ittzés, Andrew Jones, Renzo Kottmann, Robert Kulawik, Sonja Leidenberger, Päivi Lyytikäinen-Saarenmaa, Cherian Mathew, Norman Morrison, Aleksandra Nenadic, Abraham Nieva de la Hidalga, Matthias Obst, Gerard Oostermeijer, Elisabeth Paymal, Graziano Pesole, Salvatore Pinto, Axel Poigné, Francisco Quevedo Fernandez, Monica Santamaria, Hannu Saarenmaa, Gergely Sipos, Karl-Heinz Sylla, Marko Tähtinen, Saverio Vicario, Rutger Aldo Vos, Alan R. Williams, Pelin Yilmaz (2016):
BioVeL: a virtual laboratory for data analysis and modelling in biodiversity science and ecology.
BMC Ecology **16**(1)
https://doi.org/10.1186/s12898-016-0103-y

[Hardisty 2019a] Alex R Hardisty, Keping Ma, Gil Nelson, Jose Fortes (2019):
'openDS' – A New Standard for Digital Specimens and Other Natural Science Digital Object Types.
Biodiversity Information Science and Standards **3**:e37033
https://doi.org/10.3897/biss.3.37033

[Hardisty 2019b] Alex Hardisty (2019):
Provisional Data Management Plan for DiSSCo infrastructure.
Zenodo, DiSSCo Deliverable D6.6
https://doi.org/10.5281/zenodo.3532937

[Hardisty 2020] Alex Hardisty, Hannu Saarenmaa, Ana Casino, Mathias Dillen, Karsten Gödderz, Quentin Groom, Helen Hardy, Dimitris Koureas, Abraham Nieva de la Hidalga, Deborah Paul, Veljo Runnel, Xavier Vermeersch, Myriam van Walsum, Luc Willemse (2020):
Conceptual design blueprint for the DiSSCo digitization infrastructure - DELIVERABLE D8.1.
Research Ideas and Outcomes **6**:e54280.
https://doi.org/10.3897/rio.6.e54280

[Hardisty 2022] Alex Hardisty, Paul Brack, Carole Goble, Laurence Livermore, Ben Scott, Quentin Groom, Stuart Owen, Stian Soiland-Reyes (2022):
The Specimen Data Refinery: A canonical workflow framework and FAIR Digital Object approach to speeding up digital mobilisation of natural history collections.
Data Intelligence **4**(2)
https://doi.org/10.1162/dint_a_00134
https://s11.no/2022/phd/specimen-data-refinery/ (Section 5.2 on page 133)

[Harrow 2021] Jennifer Harrow, John Hancock, ELIXIR-EXCELERATE Community, Niklas Blomberg (2021):
ELIXIR-EXCELERATE: establishing Europe's data infrastructure for the life science research of the future.
EMBO Journal **40**(6):e107409
https://doi.org/10.15252/embj.2020107409

[Harrow 2022] Jennifer Harrow, Rachel Drysdale, Andrew Smith, Susanna Repo, Jerry Lanfaear, Niklas Blomberg (2022):
ELIXIR: providing a sustainable infrastructure for life science data at European scale.
Bioinformatics **37**(16)
https://doi.org/10.1093/bioinformatics/btab481

[Hashem 2015] Ibrahim Abaker Targio Hashem, Ibrar Yaqoob, Nor Badrul Anuar, Salimah Mokhtar, Abdullah Gani, Samee Ullah Khan (2015):
The rise of "big data" on cloud computing: Review and open research issues
Information Systems **47**
https://doi.org/10.1016/j.is.2014.07.006

[Hasnain 2018] Ali Hasnain, Dietrich Rebholz-Schuhmann (2019):
Assessing FAIR Data Principles Against the 5-Star Open Data Principles.
ESWC 2018: The Semantic Web: ESWC 2018 Satellite Events,
Lecture Notes in Computer Science **11155**
https://doi.org/10.1007/978-3-319-98192-5_60

[Hausenblas 2012] Michael Hausenblas, James G Kim. (2012):
5-star Open Data.
http://5stardata.info/ (accessed 24 January 2023)

[Heath 2011] Tom Heath, Christian Bizer (2011):
Linked Data: Evolving the Web into a Global Data Space.
Synthesis Lectures on the Semantic Web: Theory and Technology **1**
https://doi.org/10.2200/S00334ED1V01Y201102WBE001

[Heberling 2019] J Mason Heberling, L Alan Prather, Stephen J Tonsor (2019):
The Changing Uses of Herbarium Data in an Era of Global Change: An Overview Using Automated Content Analysis.
BioScience **69**(10)
https://doi.org/10.1093/biosci/biz094

[Heberling 2021] J Mason Heberling, Joseph T Miller, Daniel Noesgaard, Scott B Weingart, Dmitry Schigel (2021):
Data integration enables global biodiversity synthesis.
Proceedings of the National Academy of Sciences **118**(6)
https://doi.org/10.1073/pnas.2018093118

[Hellström 2022] Maggie Hellström, Carlo Zwölf, Peter Wittenburg (2022):
FDO – granularity, versioning, mutability.
FDO Specification Documents PR-Granularity-2.2-20221017
FAIR Digital Objects Forum
https://doi.org/10.5281/zenodo.7825686

[Hereld 2019] Mark Hereld, Nicola Ferrier (2019):
LightningBug ONE: An experiment in high-throughput digitization of pinned insects.
Biodiversity Information Science and Standards **3**:e37228.
https://doi.org/10.3897/biss.3.37228

[Herschel 2017] Melanie Herschel, Ralf Diestelkämper, Houssem Ben Lahmar (2017):
A survey on provenance: What for? What form? What from?
The VLDB Journal **26**
https://doi.org/10.1007/s00778-017-0486-1

[Himanen 2019] Lauri Himanen, Amber Geurts, Adam Stuart Foster, Patrick Rinke (2019):
Data-Driven Materials Science: Status, Challenges, and Perspectives.
Advanced Science **6**(21):1900808
https://doi.org/10.1002/advs.201900808

[Hitzler 2016] Pascal Hitzler, Aldo Gengami, Krzysztof Janowicz, Adila Krisnadhi, Valentina Presutti (eds.) 2016:
Ontology engineering with ontology design patterns: Foundations and Applications.
Studies on the Semantic Web **25**
ISBN 978-1-61499-676-7

[Holland 2014] Vicki Tardif Holland, Jason Johnson (2014):
Introducing 'Role'.
schema blog
http://blog.schema.org/2014/06/introducing-role.html

[Horrocks 2022] Ian Horrocks, James Hendler, eds. (2002):
The Semantic Web — ISWC 2002
First International Semantic Web Conference, Sardinia, Italy, June 9-12, 2002
https://doi.org/10.1007/3-540-48005-6

[Hospital 2020] Adam Hospital, Genís Bayarri, Stian Soiland-Reyes, Jose Lluis Gelpi, Pau Andrio, Daniele Lezzi, Sarah Butcher, Ania Niewielska, Yvonne Westermaier, Rosa Maria Badia, Rodrigo Vargas, Alexandre Bonvin (2020):
BioExcel-2 Deliverable 2.3 – First release of demonstration workflows.
Project deliverable, *Zenodo*
https://doi.org/10.5281/zenodo.4540432

[Hospital 2021a] Adam Hospital, Pau Andrio (2021):
Protein MD Setup HPC tutorial using BioExcel Building Blocks (biobb) in PyCOMPSs.
WorkflowHub. Workflow (PyCOMPSs)
https://doi.org/10.48546/workflowhub.workflow.200.1

[Hospital 2021b] Adam Hospital (2021):
Protein MD Setup tutorial using BioExcel Building Blocks (biobb) in KNIME.
WorkflowHub. Workflow (KNIME).
https://doi.org/10.48546/workflowhub.workflow.201.1

[Hu 2011] Wei Hu, Jianfeng Chen, Hang Zhang, Yuzhong Qu (2011):
How matchable are four thousand ontologies on the semantic web.
In Grigoris Antoniou, Marko Grobelnik, Elena Simperl, Bijan Parsia, Dimitris Plexousakis, Pieter De Leenheer, Jeff Pan,eds., *The semantic web: Research and applications*, pp. 290–304
ISBN 978-3-642-21033-4

[Hui 2012] Yuk Hui (2012):
What is a Digital Object?
Metaphilosophy **43**(4)
https://doi.org/10.1111/j.1467-9973.2012.01761.x

[Huntingford 2019] Chris Huntingford, Elizabeth S Jeffers, Michael B Bonsall, Hannah M Christensen, Thomas Lees and Hui Yang (2019):
Machine learning and artificial intelligence to aid climate change research and preparedness.
Environmental Research Letters **14**(12):124007
https://doi.org/10.1088/1748-9326/ab4e55

[Hussein 2021] Burhan Rashid Hussein, Owais Ahmed Malik, Wee-Hong Ong, Johan Willem Frederik Slik (2021):
Application of Computer Vision and Machine Learning for Digitized Herbarium Specimens: A Systematic Literature Review.
arXiv:2104.08732v1
https://doi.org/10.48550/arXiv.2104.08732

[IEEE 2791-2020] Raja Mazumder, Vahan Simonyan (eds.) (2020):
IEEE Standard for Bioinformatics Analyses Generated by High-Throughput Sequencing (HTS) to Facilitate Communication.
IEEE Std **2791-2020**.
ISBN 978-1-5044-6466-6.
https://research.manchester.ac.uk/en/publications/936de52b-ac53-4f0e-9927-77fd7073e88d
https://doi.org/10.1109/ieeestd.2020.9094416

[Ioannidis 2005] John P. A. Ioannidis (2005):
Why Most Published Research Findings Are False.
PLOS Medicine **19**(8):e1004085
https://doi.org/10.1371/journal.pmed.1004085

[Isaac 2009] Antoine Isaac, Ed Summers (2009):
SKOS Simple Knowledge Organization System Primer.
W3C Working Group Note 18 August 2009
https://www.w3.org/TR/2009/NOTE-skos-primer-20090818/

[Islam 2020] Sharif Islam, Alex Hardisty, Wouter Addink, Claus Weiland, Falko Glöckler (2020):
Incorporating RDA Outputs in the Design of a European Research Infrastructure for natural history Collections.
Data Science Journal **19**:50
https://doi.org/10.5334/dsj-2020-050

[Islam 2023] Sharif Islam (2023):
FAIR digital objects, persistent identifiers and machine actionability.
FAIR Connect **1**(1)
https://doi.org/10.3233/FC-230001

[ISO 16684] ISO (2019):
ISO 16684-1:2019 — graphic technology — extensible metadata platform (XMP) — part 1: Data model, serialization and core properties.
ISO standard
https://www.iso.org/standard/75163.html

[ISO 23009-1] ISO/IEC (2022):
ISO/IEC 23009-1:2022 — information technology — dynamic adaptive streaming over HTTP (DASH) — part 1: Media presentation description and segment formats.
ISO standard
https://www.iso.org/standard/83314.html

[Ison 2013] Jon Ison, Matúš Kalaš, Inge Jonassen, Dan Bolser, Mahmut Uludag, Hamish McWilliam, James Malone, Rodrigo Lopez, Steve Pettifer, Peter Rice (2013):
EDAM: an ontology of bioinformatics operations, types of data and identifiers, topics and formats.
Bioinformatics **29**(10) pp 1325–1332.
https://doi.org/10.1093/bioinformatics/btt113

[Ison 2021] Jon Ison, Hans Ienasescu, Emil Rydza, Piotr Chmura, Kristoffer Rapacki, Alban Gaignard, Veit Schwämmle, Jacques van Helden, Matúš Kalaš, Hervé Ménager (2021):
biotoolsSchema: a formalized schema for bioinformatics software description.
GigaScience **10**(1):giaa157
https://doi.org/10.1093/gigascience/giaa157

[ITU-T X.1255] ITU-T (2013):
X.1255 : Framework for Discovery of Identity Management Information.
Series X: Data networks, open system communications and security ITU-T X.1255
The International Telecommunication Union (ITU).
https://www.itu.int/rec/T-REC-X.1255-201309-I

[Iyengar 2021] Jana Iyengar, Martin Thomson (2021):
QUIC: A UDP-Based Multiplexed and Secure Transport
RFC Editor, RFC 9000
https://doi.org/10.17487/rfc9000

[Jacobsen 2020] Annika Jacobsen, Ricardo de Miranda Azevedo, Nick Juty, Dominique Batista, Simon Coles, Ronald Cornet, Mélanie Courtot, Mercè Crosas, Michel Dumontier, Chris T. Evelo, Carole Goble, Giancarlo Guizzardi, Karsten Kryger Hansen, Ali Hasnain, Kristina Hettne, Jaap Heringa, Rob W.W. Hooft, Melanie Imming, Keith G. Jeffery, Rajaram Kaliyaperumal, Martijn G. Kersloot, Christine R. Kirkpatrick, Tobias Kuhn, Ignasi Labastida, Barbara Magagna, Peter McQuilton, Natalie Meyers, Annalisa Montesanti, Mirjam van Reisen, Philippe Rocca-Serra, Robert Pergl, Susanna-Assunta Sansone, Luiz Olavo Bonino da Silva Santos, Juliane Schneider, George Strawn, Mark Thompson, Andra Waagmeester, Tobias Weigel, Mark D. Wilkinson, Egon L. Willighagen, Peter Wittenburg, Marco Roos, Barend Mons, Erik Schultes (2020):
FAIR Principles: Interpretations and Implementation Considerations.
Data Intelligence **2**(1):10–29
https://doi.org/10.1162/dint_r_00024

[Jaradeh 2019] Mohamad Yaser Jaradeh, Allard Oelen, Manuel Prinz, Markus Stocker, Sören Auer (2019):
Open research knowledge graph: A system walkthrough.
Digital libraries for open knowledge 348–351.
https://doi.org/10.1007/978-3-030-30760-8_31

[Jensen 2017] Mark A Jensen, Vincent Ferretti, Robert L Grossman, Louis M Staudt (2017):
The NCI Genomic Data Commons as an engine for precision medicine.
Blood **130**(4) pp. 453–459.
https://doi.org/10.1182/blood-2017-03-735654

[Jones 2021] Matthew B. Jones, Stephen Richard, Dave Vieglais, Adam Shepherd, Ruth Duerr, Douglas Fils, Lewis John McGibbney (2021):
Science-on-Schema.org v1.2.0
https://doi.org/10.5281/zenodo.4477164

[Jones 2022] Richard Jones, Neil Jefferies (2022):
SWORD 3.0 Specification.
https://swordapp.github.io/swordv3/swordv3.html (accessed 26 May 2022)

[Joras 2020] Matt Joras, Yang Chi (2020):
How Facebook is bringing QUIC to billions.
Engineering at Meta
https://engineering.fb.com/2020/10/21/networking-traffic/how-facebook-is-bringing-quic-to-billions

[JSONPath 2023] JSONPath WG (2023):
JSONPath: Query Expressions for JSON.
Stefan Gössner, Glyn Normington, Carsten Bormann (eds).
Internet-Draft draft-ietf-jsonpath-base-10
https://datatracker.ietf.org/doc/id/draft-ietf-jsonpath-base-10

[Jupyter 2018] Jupyter Project, Matthias Bussonnier, Jessica Forde, Jeremy Freeman, Brian Granger, Tim Head, Chris Holdgraf, Kyle Kelley, Gladys Nalvarte, Andrew Osheroff, M Pacer, Yuvi Panda, Fernando Perez, Benjamin Ragan-Kelley, Carol Willing (2018):
Binder 2.0 - Reproducible, Interactive, Sharable Environments for Science at Scale.
Proceedings of the 17th Python in Science Conference (SciPy 2018)
https://doi.org/10.25080/majora-4af1f417-011

[Juty 2011] Nick Juty, Nicolas Le Novère, Camille Laibe (2011):
Identifiers.org and MIRIAM Registry: Community resources to provide persistent identification.
Nucleic Acids Research **40**(D1)
https://doi.org/10.1093/nar/gkr1097

[Juty 2020] Nick Juty, Sarala M. Wimalaratne, Stian Soiland-Reyes, John Kunze, Carole A. Goble, Tim Clark (2020):
Unique, Persistent, Resolvable: Identifiers as the foundation of FAIR. *Data Intelligence* **2**(1):30–39
https://doi.org/10.1162/dint_a_00025

[Kahn 1995] Robert Kahn, Robert Wilensky (1995):
A framework for distributed digital object services (CNRI).
http://www.cnri.reston.va.us/k-w.html (accessed 9 May 2022)

[Kahn 2006] Robert Kahn, Robert Wilensky (2006):
A framework for distributed digital object services.
International Journal on Digital Libraries **6**(2)
https://doi.org/10.1007/s00799-005-0128-x

[Kallinikos 2013] Jannis Kallinikos, Aleksi Ville Aaltonen, Attila Marton (2013):
The ambivalent ontology of digital artifacts.
MIS Quarterly **37**(2) pp. 357–370.
ISSN 0276-7783
https://www.jstor.org/stable/43825913
https://misq.umn.edu/the-ambivalent-ontology-of-digital-artifacts.html

[Kamdar 2017] Maulik R. Kamdar, Tania Tudorache, Mark A. Musen (2017):
A systematic analysis of term reuse and term overlap across biomedical ontologies.
Semantic Web **8**(6)
https://doi.org/10.3233/sw-160238

[Katsumi 2016] Megan Katsumi, Michael Grüninger (2016):
What is ontology reuse?
In: *Formal Ontology in Information Systems*, R. Ferrario, W. Kuhn (eds.),
Frontiers in Artificial Intelligence and Applications **283**
https://doi.org/10.3233/978-1-61499-660-6-9

[Katz 2021a] Daniel S. Katz, Morane Gruenpeter, Tom Honeyman, Lorraine Hwang, Mark D. Wilkinson, Vanessa Sochat, Hartwig Anzt, Carole Goble, FAIR4RS Subgroup 1 (2021):
A Fresh Look at FAIR for Research Software.
arXiv:2101.10883 [cs.SE]
https://doi.org/10.48550/arXiv.2101.10883

[Katz 2021b] Daniel S. Katz, Morane Gruenpeter, Tom Honeyman (2021):
A Fresh Look at FAIR for Research Software.
Patterns **2**(3)
https://doi.org/10.1016/j.patter.2021.100222

[Kelly 2016] Mike Kelly (2016):
JSON Hypertext Application Language.
Internet Engineering Task Force
https://datatracker.ietf.org/doc/draft-kelly-json-hal/08/

[Khan 2019] Farah Zaib Khan, Stian Soiland-Reyes, Richard O. Sinnott, Andrew Lonie, Carole Goble, Michael R. Crusoe (2019):
Sharing interoperable workflow provenance: A review of best practices and their practical application in CWLProv.
GigaScience **8**(11)
https://doi.org/10.1093/gigascience/giz095

[Khare 2000] Rohit Khare, Scott Lawrence (2000):
Upgrading to TLS Within HTTP/1.1.
RFC Editor, RFC 2817.
https://doi.org/10.17487/rfc2817

[Kharouba 2019] Heather M. Kharouba, Jayme M. M. Lewthwaite, Rob Guralnick, Jeremy T. Kerr, Mark Vellend (2019):
Using insect natural history collections to study global change impacts: challenges and opportunities.
Philosophical Transactions of the Royal Society B: Biological Sciences **374**(1763):20170405
https://doi.org/10.1098/rstb.2017.0405

[Kim 2008] Jihie Kim, Ewa Deelman, Yolanda Gil, Gaurang Mehta, Varun Ratnakar (2008):
Provenance trails in the Wings/Pegasus system.
Concurrency and Computation: Practice and Experience **20**(5) pp. 587–597.
https://doi.org/10.1002/cpe.1228

[Kinoshita 2023] Bruno de Paula Kinoshita (2023):
RO-Crate created using Autosubmit version 4.0.100 workflow running kinow/auto-mhm-test-domains.
Zenodo
https://doi.org/10.5281/zenodo.8144612

[Kitchin 2021] Rob Kitchin (2021):
The Data Revolution: A Critical Analysis of Big Data, Open Data and Data Infrastructures. Second edition.
SAGE Publications Ltd.
ISBN 978-1-5297-3376-1

[Klein 2014] Martin Klein, Herbert Van de Sompel, Robert Sanderson, Harihar Shankar, Lyudmila Balakireva, Ke Zhou, Richard Tobin (2014):
Scholarly Context Not Found: One in Five Articles Suffers from Reference Rot.
PLOS ONE **9**(12):e115253
https://doi.org/10.1371/journal.pone.0115253

[Klímek 2019] Jakub Klímek, Petr Škoda, Martin Nečaský (2019):
Survey of tools for Linked Data consumption.
Semantic Web **10**(4)
https://doi.org/10.3233/SW-180316

[Kluyver 2016] Thomas Kluyver, Benjamin Ragan-Kelley, Fernando Pérez, Brian Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley, Jessica Hamrick, Jason Grout, Sylvain Corlay, Paul Ivanov, Damián Avila, Safia Abdalla, Carol Willing, Jupyter Development Team (2016):
Jupyter Notebooks – a publishing format for reproducible computational workflows.
Proceedings of the 20th International Conference on Electronic Publishing
Positioning and Power in Academic Publishing: Players, Agents and Agendas
https://doi.org/10.3233/978-1-61499-649-1-87

[Knyshov 2021] Alexander Knyshov, Samantha Hoang, Christiane Weirauch (2021):
Pretrained Convolutional Neural Networks Perform Well in a Challenging Test Case: Identification of Plant Bugs (Hemiptera: Miridae) Using a Small Number of Training Images.
Insect Systematics and Diversity **5**(2)
https://doi.org/10.1093/isd/ixab004

[Koesten 2020] Laura Koesten, Pavlos Vougiouklis, Elena Simperl, Paul Groth (2020):
Dataset reuse: Toward translating principles to practice.
Patterns **1**(8):100136.
https://doi.org/10.1016/j.patter.2020.100136

[Koesten 2021] Laura Koesten, Kathleen Gregory, Paul Groth, Elena Simperl (2021):
Talking datasets – understanding data sensemaking behaviours.
International journal of human-computer studies **146**:102562.
https://doi.org/10.1016/j.ijhcs.2020.102562

[Kontokostas 2017] Dimitris Kontokostas, Holger Knublauch (2017):
Shapes Constraint Language (SHACL).
W3C Recommendation
https://www.w3.org/TR/shacl/ (accessed 26 May 2022)

[Köster 2012] Johannes Köster, Sven Rahmann (2012):
Snakemake—a scalable bioinformatics workflow engine.
Bioinformatics **28**(19) pp. 2520–2522.
https://doi.org/10.1093/bioinformatics/bts480

[Kuhn 2021] Tobias Kuhn, Vincent Emonet, Haris Antonatos, Stian Soiland-Reyes, Michel Dumontier (2021):
Semantic micro-contributions with decentralized nanopublication services.
PeerJ Computer Science **7**:e387
https://doi.org/10.7717/peerj-cs.387
https://s11.no/2021/phd/nanopub/ (Supplement 5)

[Kumar 2013] Rohini Kumar, Luis Samaniego, Sabine Attinger (2013):
Implications of distributed hydrologic model parameterization on water fluxes at multiple scales and locations.
Water Resources Research **49**(1)
https://doi.org/10.1029/2012WR012195

[Kunze 2018] John A. Kunze, Justin Littman, Liz Madden, John Scancella, Chris Adams (2018):
The BagIt File Packaging Format, (V1.0)
RFC 8493
Internet Requests for Comments, RFC Editor.
https://doi.org/10.17487/rfc8493

[Kunze 2022] John A. Kunze, Emmanuelle Bermès (2022):
The ARK Identifier Scheme.
Internet Engineering Task Force
https://datatracker.ietf.org/doc/draft-kunze-ark/36/

[Kurowski 2021] Oscar Corcho, Magnus Eriksson, Krzysztof Kurowski, Milan Ojsteršek, Christine Choirat, Mark Sanden, Frederik Coppens, EOSC Executive Board (2021):
EOSC Interoperability Framework.
Publications Office of the EU, Technical Report, 2021.
https://doi.org/10.2777/620649

[Käfer 2018a] Tobias Käfer, Andreas Harth (2018):
Rule-based Programming of User Agents for Linked Data.
WWW2018 Workshop on Linked Data on the Web (LDOW2018)
http://events.linkeddata.org/ldow2018/papers/LDOW2018_paper_7.pdf

[Käfer 2018b] Tobias Käfer, Andreas Harth (2018):
Specifying, Monitoring, and Executing Workflows in Linked Data Environments.
The Semantic Web – ISWC 2018
Lecture Notes in Computer Science **11136**
arXiv:1804.05044
https://doi.org/10.1007/978-3-030-00671-6_25

[La Rosa 2021a] Marco La Rosa (2021):
GitHub – CoEDL/modpdsc.
GitHub
https://github.com/CoEDL/modpdsc/

[La Rosa 2021b] Marco La Rosa (2021):
GitHub – CoEDL/ocfl-tools: Tools to process and manipulate an OCFL tree.
https://github.com/CoEDL/ocfl-tools

[La Rosa 2021c] Marco La Rosa (2021):
Arkisto Platform: Describo Online.
https://arkisto-platform.github.io/describo-online/

[La Rosa 2021d] Marco La Rosa, Peter Sefton (2021):
Arkisto Platform: Describo.
https://arkisto-platform.github.io/describo/

[Labra Gayo 2017] Jose Emilio Labra Gayo, Eric Prud'hommeaux, Iovka Boneva, Dimitris Kontokostas (2017):
Validating RDF Data.
Synthesis Lectures on the Semantic Web: Theory and Technology **7**
https://doi.org/10.2200/s00786ed1v01y201707wbe016

[Lagoze 2008] Carl Lagoze, Herbert Van de Sompel, Pete Johnston, Michael Nelson, Robert Sanderson, Simeon Warner (2008):
ORE Specification - Abstract Data Model.
Open Archives Initiative
http://www.openarchives.org/ore/1.0/datamodel#Proxies

[Lammey 2020] R. Lammey (2020):
Solutions for identification problems: A look at the research organization registry.
Science Editing **7**(1) pp. 65–69.
https://doi.org/10.6087/kcse.192

[Lamprecht 2019] Anna-Lena Lamprecht, Leyla Garcia, Mateusz Kuzak, Carlos Martinez, Ricardo Arcila, Eva Martin Del Pico, Victoria Dominguez Del Angel, Stephanie Van De Sandt, Jon Ison, Paula Andrea Martinez, Peter Mcquilton, Alfonso Valencia, Jennifer Harrow, Fotis Psomopoulos, Josep Ll. Gelpi, Neil Chue Hong, Carole Goble, Salvador Capella-Gutierrez (2019):
Towards FAIR principles for research software.
Data Science **3**(1) pp. 37–59
https://doi.org/10.3233/DS-190026

[Lamprecht 2021] Anna-Lena Lamprecht, Magnus Palmblad, Jon Ison, Veit Schwämmle, Mohammad Sadnan Al Manir, Ilkay Altintas, Christopher J. O. Baker, Ammar Ben Hadj Amor, Salvador Capella-Gutierrez, Paulos Charonyktakis, Michael R. Crusoe, Yolanda Gil, Carole Goble, Timothy J. Griffin, Paul Groth, Hans Ienasescu, Pratik Jagtap, Matúš Kalaš, Vedran Kasalica, Alireza Khanteymoori, Tobias Kuhn, Hailiang Mei, Hervé Ménager, Steffen Möller, Robin A. Richardson, Vincent Robert, Stian Soiland-Reyes, Robert Stevens, Szoke Szaniszlo, Suzan Verberne, Aswin Verhoeven, Katherine Wolstencroft (2021):
Perspectives on automated composition of workflows in the life sciences.
F1000Research **10** (2021):897
https://doi.org/10.12688/f1000research.54159.1

[Lannom 2020] Larry Lannom, Dimitris Koureas, Alex R. Hardisty (2020):
FAIR Data and Services in Biodiversity Science and Geoscience.
Data Intelligence **2**(1–2):122–130.
https://doi.org/10.1162/dint_a_00034

[Lannom 2022a] Larry Lannom, Karsten Peters-von Gehlen, Ivonne Anders, Andreas Pfeil, Alexander Schlemmer, Zach Trautt, Peter Wittenburg (2022):
FDO configuration types.
FDO Specification Documents PR-ConfigurationTypes-2.1-20221017
FAIR Digital Objects Forum
https://doi.org/10.5281/zenodo.7825703

[Lannom 2022b] Larry Lannom, Ulrich Schwardmann, Christophe Blanchi, Ivonne Anders, Claus Weiland, Peter Wittenburg (2022):
FAIR digital objects roadmap. Version 5 november 2022.
FAIR Digital Objects Forum
https://doi.org/10.5281/zenodo.7824673

[Lannom 2022c] Larry Lannom, Ulrich Schwardmann, Cristophe Blanchi, Peter Wittenburg (2022):
Typing FAIR digital objects.
FDO Specification Documents PR-TypingFDOs-2.0-20220608
FAIR Digital Objects Forum
https://doi.org/10.5281/zenodo.7825599

[Lanthaler 2021] Markus Lanthaler, ed. (2021):
Hydra Core Vocabulary.
Hydra W3C Community Group
http://www.hydra-cg.com/spec/latest/core/

[Lassila 1999] Ora Lassila, Ralph R. Swick (1999):
Resource Description Framework (RDF) Model and Syntax Specification.
W3C Recommendation
https://www.w3.org/TR/1999/REC-rdf-syntax-19990222/

[Lebo 2013a] Timothy Lebo, Satya Sahoo, Deborah McGuinness, Khalid Belhajjame, James Cheney, David Corsar, Daniel Garijo, Stian Soiland-Reyes, Stephan Zednik, Jun Zhao (2013):
PROV-O: The PROV Ontology.
W3C Recommendation 30 April 2013.
https://www.w3.org/TR/2013/REC-prov-o-20130430/

[Lebo 2013b] Timothy Lebo, Luc Moreau (2013):
Linking Across Provenance Bundles.
W3C Working Group Note 30 April 2013
https://www.w3.org/TR/2013/NOTE-prov-links-20130430/

[Leach 2005] Paul J. Leach, Rich Salz, Michael H. Mealling (2005):
A Universally Unique IDentifier (UUID) URN Namespace.
RFC Editor, RFC 4122
https://doi.org/10.17487/rfc4122

[Lee 2018] Benjamin D. Lee (2018):
Ten simple rules for documenting scientific software.
PLOS Computational Biology **14**(12):e1006561
https://doi.org/10.1371/journal.pcbi.1006561

[Leipzig 2021] Jeremy Leipzig, Daniel Nüst, Charles Tapley Hoyt, Karthik Ram, Jane Greenberg (2021):
The role of metadata in reproducible computational research.
Patterns **2**(9):100322.
https://doi.org/10.1016/j.patter.2021.100322

[Leo 2023a] Simone Leo, Stian Soiland-Reyes, Michael R. Crusoe (2023):
runcrate 0.5.0
Zenodo / GitHub
https://github.com/ResearchObject/runcrate
https://doi.org/10.5281/zenodo.10203433

[Leo 2023b] Simone Leo, Michael R. Crusoe, Laura Rodríguez-Navas, Raül Sirvent, Alexander Kanitz, Paul De Geest, Rudolf Wittner, Luca Pireddu, Daniel Garijo, José M. Fernández, Iacopo Colonnelli, Matej Gallo, Tazro Ohta, Hirotaka Suetake, Salvador Capella-Gutierrez, Renske de Wit, Bruno de Paula Kinoshita, Stian Soiland-Reyes (2023):
Recording provenance of workflow runs with RO-Crate (RO-Crate and mapping).
RO-Crate
Zenodo
https://w3id.org/ro/doi/10.5281/zenodo.10368989
https://doi.org/10.5281/zenodo.10368989

[Leo 2023c] Simone Leo (2023):
Run of digital pathology tissue/tumor prediction workflow.
Zenodo
https://doi.org/10.5281/zenodo.7774351

[Leo 2024] Simone Leo, Michael R. Crusoe, Laura Rodríguez-Navas, Raül Sirvent, Alexander Kanitz, Paul De Geest, Rudolf Wittner, Luca Pireddu, Daniel Garijo, José M. Fernández, Iacopo Colonnelli, Matej Gallo, Tazro Ohta, Hirotaka Suetake, Salvador Capella-Gutierrez, Renske de Wit, Bruno de Paula Kinoshita, Stian Soiland-Reyes (2024):
Recording provenance of workflow runs with RO-Crate.
PLOS One **19**(9):e0309210
arXiv 2312.07852 [cs.DL]
https://doi.org/10.48550/arXiv.2312.07852
https://doi.org/10.1371/journal.pone.0309210
https://s11.no/2023/phd/workflow-run-crate/ (Section 5.4 on page 152)

[Little 2020] Damon P. Little, Melissa Tulig, Kiat Chuan Tan, Yulong Liu, Serge Belongie, Christine Kaeser-Chen, Fabián A. Michelangeli, Kiran Panesar, R.V. Guha, Barbara A. Ambrose (2020):
An algorithm competition for automatic species identification from herbarium specimens.
Applications in Plant Sciences **8**(6):e11365
https://doi.org/10.1002/aps3.11365

[Liu 2007] Kevin Liu, David Booth (2007):
Web Services Description Language (WSDL) Version 2.0 Part 0: Primer.
W3C Recommendation 26 June 2007
https://www.w3.org/TR/2007/REC-wsdl20-primer-20070626/

[Livermore 2022a] Laurence Livermore, Oliver Woolland (2022):
DLA-Collections-test. (Galaxy workflow)
WorkflowHub
https://doi.org/10.48546/workflowhub.workflow.374.1

[Livermore 2022b] Laurence Livermore, Oliver Woolland (2022):
HTR-Collections-test. (Galaxy workflow)
WorkflowHub
https://doi.org/10.48546/workflowhub.workflow.375.1

[Lohonya 2020] Krisztina Lohonya, Laurence Livermore, Malcolm Penn (2020):
Georeferencing the Natural History Museum's Chinese type collection: of plateaus, pagodas and plants.
Biodiversity Data Journal **8**:e50503.
https://doi.org/10.3897/BDJ.8.e50503

[Loo 2022] Tina Loo, ed. (2022):
First International Conference on FAIR Digital Objects.
Research Ideas and Outcomes
https://doi.org/10.3897/rio.coll.190

[Lordan 2014] Francesc Lordan, Enric Tejedor, Jorge Ejarque, Roger Rafanell, Javier Álvarez, Fabrizio Marozzo, Daniele Lezzi, Raül Sirvent, Domenico Talia, Rosa M. Badia (2014):
ServiceSs: An interoperable programming framework for the cloud.
Journal of Grid Computing, **12**(1)
https://doi.org/10.1007/s10723-013-9272-5

[Lowe 2021a] Douglas Lowe (2021):
Protein MD Setup tutorial using BioExcel Building Blocks (biobb) in Galaxy.
WorkflowHub. Workflow (Galaxy).
https://doi.org/10.48546/workflowhub.workflow.194.1

[Lowe 2021b] Douglas Lowe, Genís Bayarri (2021):
Protein Ligand Complex MD Setup tutorial using BioExcel Building Blocks (biobb) (jupyter notebook).
https://doi.org/10.48546/workflowhub.workflow.56.1

[Ludäscher 2016] Bertram Ludäscher (2016):
A Brief Tour Through Provenance in Scientific Workflows and Databases.
Building Trust in Information, Springer Proceedings in Business and Economics
https://doi.org/10.1007/978-3-319-40226-0_7

[Lughadha 2019] Eimear M. Nic Lughadha, Vanessa Graziele Staggemeier, Thais N. C. Vasconcelos, Barnaby E. Walker, Cátia Canteiro, Eve J. Lucas (2019):
Harnessing the potential of integrated systematics for conservation of taxonomically complex, megadiverse plant groups.
Conservation Biology **33**(3)
https://doi.org/10.1111/cobi.13289

[Luo 2022] Yu Luo (2022):
Knowledge enhanced digital objects.
Doctoral dissertation. Indiana University.
ISBN 9798368418988
https://www.proquest.com/docview/2763290077

[Luo 2023] Yu Luo, Beth Plale (2023):
Knowledge Enhanced Digital Objects: a Data Lake Approach.
IEEE/ACM 23rd International Symposium on Cluster, Cloud and Internet Computing Workshops (*CCGridW*)
https://doi.org/10.1109/CCGridW59191.2023.00064

[Lynch 2022] Mike Lynch, Peter Sefton (2022):
npm: ro-crate-excel.
npm
https://www.npmjs.com/package/ro-crate-excel

[Mai Chan 1995] Lois Mai Chan (1995):
Library of Congress Subject Headings: Principles and Application, 3rd edition.
ISBN 9781563081910.

[Manubens-Gil 2016] Domingo Manubens-Gil, Javier Vegas-Regidor, Chloe Prodhomme, Oriol Mula-Valls, Francisco J. Doblas-Reyes (2016):
Seamless management of ensemble climate prediction experiments on HPC platforms.
2016 International Conference on High Performance Computing & Simulation (HPCS), Innsbruck, Austria
https://doi.org/10.1109/HPCSim.2016.7568429

[Marinescu 2023] Dan C. Marinescu (2022):
Cloud Computing: Theory and Practice. Third edition.
Morgan Kaufmann Publishers
ISBN 978-0-323-85277-7

[Matentzoglu 2022] Nicolas Matentzoglu, Damien Goutte-Gattat, Shawn Zheng Kai Tan, James P Balhoff, Seth Carbon, Anita R Caron, William D Duncan, Joe E Flack, Melissa Haendel, Nomi L Harris, William R Hogan, Charles Tapley Hoyt, Rebecca C Jackson, HyeongSik Kim, Huseyin Kir, Martin Larralde, Julie A McMurry, James A Overton, Bjoern Peters, Clare Pilgrim, Ray Stefancsik, Sofia MC Robb, Sabrina Toro, Nicole A Vasilevsky, Ramona Walls, Christopher J Mungall, David Osumi-Sutherland (2022):
Ontology Development Kit: a toolkit for building, maintaining and standardizing biomedical ontologies.
Database **2022**:baac087
https://doi.org/10.1093/database/baac087

[McMurry 2017] Julie A McMurry, Nick Juty, Niklas Blomberg, Tony Burdett, Tom Conlin, Nathalie Conte, Mélanie Courtot, John Deck, Michel Dumontier, Donal K Fellows, Alejandra Gonzalez-Beltran, Philipp Gormanns, Jeffrey Grethe, Janna Hastings, Jean-Karim Hériché, Henning Hermjakob, Jon C Ison, Rafael C Jimenez, Simon Jupp, John Kunze, Camille Laibe, Nicolas Le Novère, James Malone, Maria Jesus Martin, Johanna R McEntyre, Chris Morris, Juha Muilu, Wolfgang Müller, Philippe Rocca-Serra, Susanna-Assunta Sansone, Murat Sariyar, Jacky L Snoep, Stian Soiland-Reyes, Natalie J Stanford, Neil Swainston, Nicole Washington, Alan R Williams, Sarala M Wimalaratne, Lilly M Winfree, Katherine Wolstencroft, Carole Goble, Cristopher J Mungall, Melissa A Haendel, Helen Parkinson (2017):
Identifiers for the 21st century: How to design, provision, and reuse identifiers to maximize utility and impact of life science data.
PLOS Biology **15**(6):e2001414
https://doi.org/10.1371/journal.pbio.2001414

[MDN 2023] MDN (2023):
HTTP Content negotiation.
Web technology for developers
MDN Web Docs
https://developer.mozilla.org/en-US/docs/Web/HTTP/Content_negotiation (accessed 26 May 2022)

[de Mello 2022] Blanda Helena de Mello, Sandro José Rigo, Cristiano André da Costa, Rodrigo da Rosa Righi, Bruna Donida, Marta Rosecler Bez, Luana Carina Schunke (2022):
Semantic interoperability in health records standards: a systematic literature review.
Health and Technology **12**
https://doi.org/10.1007/s12553-022-00639-w

[Meroño-Peñuela 2021a] Albert Meroño-Peñuela, Pasquale Lisena, Carlos Martínez-Ortiz (2021):
Conclusion and future challenges.
Web data apis for knowledge graphs: Easing access to semantic data for application developers
Synthesis Lectures on Data, Semantics, and Knowledge
https://doi.org/10.1007/978-3-031-01917-3_7

[Meroño-Peñuela 2021b] Albert Meroño-Peñuela, Pasquale Lisena, Carlos Martínez-Ortiz (2021):
Web data APIs over SPARQL.
Web Data APIs for Knowledge Graphs : Easing access to semantic data for application developers, Synthesis Lectures on Data, Semantics, and Knowledge.
https://doi.org/10.1007/978-3-031-01917-3_3

[Meurisse 2023] Marjan Meurisse, Francisco Estupiñán-Romero, Javier González-Galindo, Natalia Martínez-Lizaga, Santiago Royo-Sierra, Simon Saldner, Lorenz Dolanski-Aghamanoukjan, Alexander Degelsegger-Marquez, Stian Soiland-Reyes, Nina Van Goethem, Enrique Bernal-Delgado, On Behalf of BeYond-COVID project contributors (2023):
Federated causal inference based on real-world observational data sources: application to a SARS-CoV-2 vaccine effectiveness assessment.
BMC Medical Research Methodology **23**:248
https://doi.org/10.1186/s12874-023-02068-3
https://s11.no/2023/phd/federated-causal-inference/ (Supplement 16)

[Miksa 2019a] Tomasz Miksa, Paul Walk, Peter Neish (2019):
RDA DMP Common Standard for Machine-Actionable Data Management Plans.
Research Data Alliance
https://doi.org/10.15497/rda00039

[Miksa 2019b] Tomasz Miksa, Stephanie Simms, Daniel Mietchen, Sarah Jones (2019):
Ten principles for machine-actionable data management plans.
PLOS Computational Biology **15**(3):e1006750
https://doi.org/10.1371/journal.pcbi.1006750

[Miksa 2020] Tomasz Miksa, Maroua Jaoua, Ghaith Arfaoui (2020):
Research Object Crates and Machine-actionable Data Management Plans.
First Workshop on Data and Research Objects Management for Linked Open Science (*DaMaLOS*) at The 19th International Semantic Web Conference (ISWC 2020).
https://doi.org/10.4126/frl01-006423291

[Millard 2010] Ian C. Millard, Hugh Glaser, Manuel Salvadores, Nigel Shadbolt (2010):
Consuming multiple linked data sources: Challenges and Experiences.
Proceedings of the First International Workshop on Consuming Linked Data (*COLD2010*), Olaf Hartig, Andreas Harth, Juan Sequeda (eds.)
CEUR Workshop Proceedings **665**
https://ceur-ws.org/Vol-665/MillardEtAl_COLD2010.pdf

[Miller 2021] Darrel Miller, Jeremy Whitlock, Marsh Gardiner, Mike Ralphson, Ron Ratovsky, Uri Sarid, eds. (2021):
OpenAPI Specification v3.1.0.
OpenAPI Initiative, The Linux Foundation.
https://spec.openapis.org/oas/v3.1.0.html (accessed 2023-11-06)

[Missier 2010] Paolo Missier, Bertram Ludascher, Shawn Bowers, Saumen Dey, Anandarup Sarkar, Biva Shrestha, Ilkay Altintas, Manish Kumar Anand, Carole Goble (2010):
Linking multiple workflow provenance traces for interoperable collaborative science.
The 5th Workshop on Workflows in Support of Large-Scale Science
https://doi.org/10.1109/WORKS.2010.5671861

[Missier 2013] Paolo Missier, Saumen Dey, Khalid Belhajjame, Víctor Cuevas-Vicenttín, Bertram Ludäscher (2013):
D-PROV: extending the PROV provenance model with workflow structure.
Proceedings of the 5th USENIX Workshop on the Theory and Practice of Provenance (*TaPP '13*)
https://doi.org/10.5555/2482949.2482961

[Möller 2010] Steffen Möller, Hajo Nils Krabbenhöft, Andreas Tille, David Paleino, Alan Williams, Katy Wolstencroft, Carole Goble, Richard Holland, Dominique Belhachemi, Charles Plessy (2010):
Community-driven computational biology with Debian Linux.
BMC Bioinformatics **11**(Suppl 12):S5
https://doi.org/10.1186/1471-2105-11-S12-S5

[Möller 2017] Steffen Möller, Stuart W. Prescott, Lars Wirzenius, Petter Reinholdtsen, Brad Chapman, Pjotr Prins, Stian Soiland-Reyes, Fabian Klötzl, Andrea Bagnacani, Matúš Kalaš, Andreas Tille, Michael R. Crusoe (2017):
Robust cross-platform workflows: How technical and scientific communities collaborate to develop, test and share best practices for data analysis.
Data Science and Engineering **2**
https://doi.org/10.1007/s41019-017-0050-4

[Mons 2017] Barend Mons, Cameron Neylon, Jan Velterop, Michel Dumontier, Luiz Olavo Bonino da Silva Santos, Mark D. Wilkinson (2017):
Cloudy, increasingly FAIR; revisiting the FAIR Data guiding principles for the European Open Science Cloud.
Information Services & Use **37**(1)
https://doi.org/10.3233/ISU-170824

[Mons 2018] Barend Mons (2018):
Data Stewardship for Open Science.
ISBN 9781315351148

[Mons 2020] Barend Mons, Erik Schultes, Fenghong Liu, Annika Jacobsen (2020):
The FAIR Principles: First Generation Implementation Choices and Challenges.
Data Intelligence **2**(1–2)
https://doi.org/10.1162/dint_e_00023

[Moreau 2013] Luc Moreau, Paolo Missier, Khalid Belhajjame, Reza B'Far, James Cheney, Sam Coppens, Stephen Cresswell, Yolanda Gil, Paul Groth, Graham Klyne, Timothy Lebo, Jim McCusker, Simon Miles, James Myers, Satya Sahoo, Curt Tilmes (2013):
PROV-DM: The PROV Data Model.
W3C Recommendation 30 April 2013
https://www.w3.org/TR/2013/REC-prov-dm-20130430/.

[myExperiment 2009] myExperiment (2009):
myExperiment Ontology Modules.
myExperiment / Internet Archive
https://web.archive.org/web/20091115080336/http%3a%2f%2frdf.myexperiment.org/ontologies

[Nature 2019] Nature Editorial (2019):
Giving software its due.
Nature Methods **16**(3)
https://doi.org/10.1038/s41592-019-0350-x

[NCBO] NCBO BioPortal.
National Center for Biomedical Ontology
https://bioportal.bioontology.org/ontologies (accessed 26 May 2022)

[Nelson 2019a] Gil Nelson, Shari Ellis (2019):
The history and impact of digitization and digital data mobilization on biodiversity research.
Philosophical Transactions of the Royal Society B: Biological Sciences **374**(1763):20170391
https://doi.org/10.1098/rstb.2017.0391

[Nelson 2019b] Gil Nelson, Deborah L Paul (2019):
DiSSCo, iDigBio and the Future of Global Collaboration.
Biodiversity Information Science and Standards **3**:e37896.
https://doi.org/10.3897/biss.3.37896

[Neumann 2021] Andy Neumann, Nuno Laranjeiro, Jorge Bernardino (2021):
An analysis of public REST web service apis.
IEEE Transactions on Services Computing **14**(4)
https://doi.org/10.1109/TSC.2018.2847344

[Newman 2009] David R Newman, Sean Bechhofer, David De Roure (2009):
myExperiment: An ontology for e-Research.
Proceedings of the Workshop on Semantic Web Applications in Scientific Discourse (SWASD 2009)
CEUR Workshop Proceedings **523**
http://ceur-ws.org/Vol-523/Newman.pdf

[Neylon 2017] Cameron Neylon (2017):
As a researcher ... I'm a bit bloody fed up with Data Management.
Science in the Open (blog)
https://cameronneylon.net/blog/as-a-researcher-im-a-bit-bloody-fed-up-with-data-management/
(accessed 2022-01-27)

[Nieva de la Hidalga 2021] Abraham Nieva de la Hidalga, Paul L. Rosin, Xianfang Sun, Laurence Livermore, James Durrant, James Turner, Mathias Dillen, Alicia Musson, Sarah Phillips, Quentin Groom, Alex Hardisty (2022):
Cross-validation of a semantic segmentation network for natural history collection specimens.
Machine Vision and Applications **33**(3)
https://doi.org/10.1007/s00138-022-01276-z

[Niewielska 2020] Ania Niewielska, Sarah Butcher, Yvonne Westermaier (2020):
BioExcel-2 Deliverable 2.5 - Provision of a Workflow Environment at BioExcel portal.
Zenodo
https://doi.org/10.5281/zenodo.4916060

[Norris 2021] Emma Norris, Janna Hastings, Marta M. Marques, Ailbhe N. Finnerty Mutlu, Silje Zink, Susan Michie (2021):
Why and how to engage expert stakeholders in ontology development: Insights from social and behavioural sciences.
Journal of Biomedical Semantics **12**(1)
https://doi.org/10.1186/s13326-021-00240-6

[Nottingham 2017] Mark Nottingham (2017):
Web Linking.
RFC Editor, RFC 8288
https://doi.org/10.17487/rfc8288

[Nurdiati 2008] Sri Nurdiati, Cornelis Hoede (2008):
25 years development of knowledge graph theory: the results and the challenge.
Memorandum No. 2/1876
University of Twente
https://purl.utwente.nl/publications/64931

[Ó Carragáin 2019a] Eoghan Ó Carragáin, Carole Goble, Peter Sefton, Stian Soiland-Reyes (2019):
A lightweight approach to research object data packaging.
Bioinformatics Open Source Conference (*BOSC2019*), 2019-07-24/2019-07-25, Basel, Switzerland.
https://doi.org/10.5281/zenodo.3250687

[Ó Carragáin 2019b] Eoghan Ó Carragáin, Carole Goble, Peter Sefton, Stian Soiland-Reyes (2019):
RO-Crate: A lightweight approach to research object data packaging.
RO-15 at *Workshop on Research Objects* (RO 2019), IEEE eScience 2019, 2019-09-24, San Diego, CA, USA.
https://doi.org/10.5281/zenodo.3337883
https://s11.no/2019/phd/ro-crate/ (Supplement 15)

[OCFL 2020] **OCFL, Oxford Common File Layout Specification**, Recommendation, 2020.
https://ocfl.io/1.0/spec/

[OCLC 2010] OCLC (2010):
"Info" URI Registry (Frozen).
OCLC
http://info-uri.info/ (accessed 2023-11-06)

[Ohta 2023] Tazro Ohta, Hirotaka Suetake (2023):
Example of Workflow Run RO-Crate Output in Sapporo
Zenodo
https://doi.org/10.5281/zenodo.10134581

[Oliver 2019] Hilary Oliver, Matthew Shin, David Matthews, Oliver Sanders, Sadie Bartholomew, Andrew Clark, Ben Fitzpatrick, Ronald van Haren, Rolf Hut, Niels Drost (2019):
Workflow Automation for Cycling Systems.
Computing in Science & Engineering **21**(4)
https://doi.org/10.1109/MCSE.2019.2906593

[Open Graph] **The Open Graph protocol**.
https://ogp.me/ (accessed 26 May 2022)

[openDS 2021] openDS (2021):
Draft specification for open Digital Specimens (openDS)
https://github.com/DiSSCo/openDS (accessed 2021-08-10)

[OpenStand 2017] OpenStand (2017):
The Modern Standards Paradigm - Five Key Principles.
https://open-stand.org/about-us/principles/ (accessed 24 January 2023)

[OSI 022] OSI (2022):
Licenses & Standards. Open Source Initiative
https://opensource.org/licenses (accessed 24 January 2023)

[Owen 2020] David Owen, Quentin Groom, Alex Hardisty, Thijs Leegwater, Laurence Livermore, Myriam van Walsum, Noortje Wijkamp, Irena Spasić (2020):
Towards a scientific workflow featuring Natural Language Processing for the digitisation of natural history collections.
Research Ideas and Outcomes **6**:e58030.
https://doi.org/10.3897/rio.6.e58030

[Page 2011] Kevin R. Page, David C. De Roure, Kirk Martinez (2011):
REST and Linked Data.
Proceedings of the Second International Workshop on RESTful Design - WS-REST '11
https://doi.org/10.1145/1967428.1967435

[Pantos 2017] Roger Pantos, William May (2017):
HTTP Live Streaming.
RFC Editor, RFC 8216
https://doi.org/10.17487/rfc8216

[Parecki 2017] Aaron Parecki, ed. (2017):
Micropub. *W3C Recommendation* 23 May 2017
https://www.w3.org/TR/2017/REC-micropub-20170523/

[Pavel 2023] Antonia Floriana Pavel, Daniel Garijo (2023):
Ya2ro: A tool for creating Research Objects from minimum metadata.
Workshop on Metadata and Research (objects) Management for Linked Open Science (3. : 2023 : Online)
https://doi.org/10.4126/frl01-006444984

[Pérez 2018] Beatriz Pérez, Julio Rubio, Carlos Sáenz-Adán (2018):
A systematic review of provenance systems.
Knowledge and Information Systems **57**
https://doi.org/10.1007/s10115-018-1164-3

[Pergl 2019] Robert Pergl, Rob Hooft, Marek Suchánek, Vojtěch Knaisl, Jan Slifka (2019):
"Data Stewardship Wizard": A Tool Bringing Together Researchers, Data Stewards, and Data Experts around Data Management Planning.
Data Science Journal **18**(1)
https://doi.org/10.5334/dsj-2019-059

[Piper 2020] Alana Piper (2020):
Digital crowdsourcing and public understandings of the past: Citizen historians meet criminal characters.
History Australia **17**(3)
https://doi.org/10.1080/14490854.2020.1796500

[Poiata 2016] Natalia Poiata, Claudio Satriano, Jean-Pierre Vilotte, Pascal Bernard, Kazushige Obara (2016):
Multiband array detection and location of seismic sources recorded by dense seismic networks.
Geophysical Journal International **205**(3)
https://doi.org/10.1093/gji/ggw071

[Poiata 2023] Natalia Poiata, Claudio Satriano, Javier Conejero (2023):
BackTrackBB: Multi-band array detection and location of seismic sources (PyCOMPSs implementation).
Zenodo
https://doi.org/10.5281/zenodo.7788030

[Polleres 2020] Axel Polleres, Maulik Rajendra Kamdar, Javier David Fernández, Tania Tudorache, Mark Alan Musen (2020):
A more decentralized vision for linked data.
Semantic Web **11**(1)
https://doi.org/10.3233/SW-190380

[Poveda 2010] María Poveda, Mari Carmen Suárez-Figueroa, Asunción Gómez-Pérez (2010):
Common Pitfalls in Ontology Development.
CAEPIA 2009: Current Topics in Artificial Intelligence.
Lecture Notes in Computer Science **5988**
https://doi.org/10.1007/978-3-642-14264-2_10

[Price 2018] Benjamin Wills Price, Steen Dupont, Elizabeth Louise Allan, Vladimir Blagoderov, Alice Jenny Butcher, James Durrant, Pieter Holtzhausen, Phaedra Kokkini, Laurence Livermore, Helen Hardy, Vincent Smith (2018):
ALICE: Angled Label Image Capture and Extraction for high throughput insect specimen digitisation.
https://doi.org/10.31219/osf.io/s2p73

[Prlić 2012] Andreas Prlić, James B. Procter (2012):
Ten Simple Rules for the Open Development of Scientific Software.
PLOS Computational Biology **8**(12)
https://doi.org/10.1371/journal.pcbi.1002802

[Pryer 2022] Kathleen M. Pryer, Carlo Tomasi, Xiaohan Wang, Emily K. Meineke, Michael D. Windham (2020):
Using computer vision on herbarium specimen images to discriminate among closely related horsetails (Equisetum).
Applications in Plant Sciences **8**(6):e11372
https://doi.org/10.1002/aps3.11372

[RDF 1.1 2014] RDF Working Group (2014):
RDF 1.1 Concepts and Abstract Syntax.
W3C Recommendation 25 Feb 2014.
https://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/

[Rehm 2021] Heidi L. Rehm, Angela J.H. Page, Lindsay Smith, Jeremy B. Adams, Gil Alterovitz, Lawrence J. Babb, Maxmillian P. Barkley, Michael Baudis, Michael J.S. Beauvais, Tim Beck, Jacques S. Beckmann, Sergi Beltran, David Bernick, Alexander Bernier, James K. Bonfield, Tiffany F. Boughtwood, Guillaume Bourque, Sarion R. Bowers, Anthony J. Brookes, Michael Brudno, Matthew H. Brush, David Bujold, Tony Burdett, Orion J. Buske, Moran N. Cabili, Daniel L. Cameron, Robert J. Carroll, Esmeralda Casas-Silva, Debyani Chakravarty, Bimal P. Chaudhari, Shu Hui Chen, J. Michael Cherry, Justina Chung, Melissa Cline, Hayley L. Clissold, Robert M. Cook-Deegan, Mélanie Courtot, Fiona Cunningham, Miro Cupak, Robert M. Davies, Danielle Denisko, Megan J. Doerr, Lena I. Dolman, Edward S. Dove, L. Jonathan Dursi, Stephanie O.M. Dyke, James A. Eddy, Karen Eilbeck, Kyle P. Ellrott, Susan Fairley, Khalid A. Fakhro, Helen V. Firth, Michael S. Fitzsimons, Marc Fiume, Paul Flicek, Ian M. Fore, Mallory A. Freeberg, Robert R. Freimuth, Lauren A. Fromont, Jonathan Fuerth, Clara L. Gaff, Weiniu Gan, Elena M. Ghanaim, David Glazer, Robert C. Green, Malachi Griffith, Obi L. Griffith, Robert L. Grossman, Tudor Groza, Jaime M. Guidry Auvil, Roderic Guigó, Dipayan Gupta, Melissa A. Haendel, Ada Hamosh, David P. Hansen, Reece K. Hart, Dean Mitchell Hartley, David Haussler, Rachele M. Hendricks-Sturrup, Calvin W.L. Ho, Ashley E. Hobb, Michael M. Hoffman, Oliver M. Hofmann, Petr Holub, Jacob Shujui Hsu, Jean-Pierre Hubaux, Sarah E. Hunt, Ammar Husami, Julius O. Jacobsen, Saumya S. Jamuar, Elizabeth L. Janes, Francis Jeanson, Aina Jené, Amber L. Johns, Yann Joly, Steven J.M. Jones, Alexander Kanitz, Kazuto Kato, Thomas M. Keane, Kristina Kekesi-Lafrance, Jerome Kelleher, Giselle Kerry, Seik-Soon Khor, Bartha M. Knoppers, Melissa A. Konopko, Kenjiro Kosaki, Martin Kuba, Jonathan Lawson, Rasko Leinonen, Stephanie Li, Michael F. Lin, Mikael Linden, Xianglin Liu, Isuru Udara Liyanage, Javier Lopez, Anneke M. Lucassen, Michael Lukowski, Alice L. Mann, John Marshall, Michele Mattioni, Alejandro Metke-Jimenez, Anna Middleton, Richard J. Milne, Fruzsina Molnár-Gábor, Nicola Mulder, Monica C. Munoz-Torres, Rishi Nag, Hidewaki Nakagawa, Jamal Nasir, Arcadi Navarro, Tristan H. Nelson, Ania Niewielska, Amy Nisselle, Jeffrey Niu, Tommi H. Nyrönen, Brian D. O'Connor, Sabine Oesterle, Soichi Ogishima, Vivian Ota Wang, Laura A.D. Paglione, Emilio Palumbo, Helen E. Parkinson, Anthony A. Philippakis, Angel D. Pizarro, Andreas Prlic, Jordi Rambla, Augusto Rendon, Renee A. Rider, Peter N. Robinson, Kurt W. Rodarmer, Laura Lyman Rodriguez, Alan F. Rubin, Manuel Rueda, Gregory A. Rushton, Rosalyn S. Ryan, Gary I. Saunders, Helen Schuilenburg, Torsten Schwede, Serena Scollen, Alexander Senf, Nathan C. Sheffield, Neerjah Skantharajah, Albert V. Smith, Heidi J. Sofia, Dylan Spalding, Amanda B. Spurdle, Zornitza Stark, Lincoln D. Stein, Makoto Suematsu, Patrick Tan, Jonathan A. Tedds, Alastair A. Thomson, Adrian Thorogood, Timothy L. Tickle, Katsushi Tokunaga, Juha Törnroos, David Torrents, Sean Upchurch, Alfonso Valencia, Roman Valls Guimera, Jessica Vamathevan, Susheel Varma, Danya F. Vears, Coby Viner, Craig Voisin, Alex H. Wagner, Susan E. Wallace, Brian P. Walsh, Marc S. Williams, Eva C. Winkler, Barbara J. Wold, Grant M. Wood, J. Patrick Woolley, Chisato Yamasaki, Andrew D. Yates, Christina K. Yung, Lyndon J. Zass, Ksenia Zaytseva, Junjun Zhang, Peter Goodhand, Kathryn North, Ewan Birney (2021):
GA4GH: International policies and standards for data sharing across genomic research and healthcare.
Cell Genomics **1**(2):100029
https://doi.org/10.1016/j.xgen.2021.100029

[Reilly 2009] Sean Reilly (2009):
Digital Object Interface Protocol Version 1.0.
https://www.dona.net/doipv1doc (accessed 26 May 2022)

[Reis 2022] David Reis, Bruno Piedade, Filipe F, Correia, João Pedro Dias, Ademar Aguiar (2022):
Developing Docker and Docker-Compose Specifications: A Developers' Survey.
IEEE Access **10** pp. 2318–2329
https://doi.org/10.1109/ACCESS.2021.3137671

[Rescorla 2000] Eric Rescorla (2000):
HTTP Over TLS.
RFC Editor, RFC 2818
https://doi.org/10.17487/rfc2818

[Rettberg 2015] Najla Rettberg, Birgit Schmidt (2015):
OpenAIRE: Supporting a European open access mandate.
College & Research Libraries News **76**(6)
https://doi.org/10.5860/crln.76.6.9326

[Riccardi 2022] Demian Riccardi, Zachary Trautt, Ala Bazyleva, Eugene Paulechka, Vladimir Diky, Joseph W. Magee, Andrei F. Kazakov, Scott A. Townsend, Chris D. Muzny (2022):
Towards improved FAIRness of the ThermoML Archive.
Journal of Computational Chemistry **43**(12)
https://doi.org/10.1002/jcc.26842

[Rice 2022] Adam Rice, Ian Hickson, Anne van Kesteren, Yutaka Hirano (2022):
WebSockets Standard.
WHATWG
https://websockets.spec.whatwg.org/ (accessed 26 May 2022)

[Riungu-Kalliosaari 2022] Leah Riungu-Kalliosaari, Rob Hooft, Sylvia Kuijpers, Jessica Parland-von Essen, Hanna Koivula (2022):
D2.10 3rd Report on FAIR requirements for persistence and interoperability.
FAIRsFAIR project deliverable
https://doi.org/10.5281/zenodo.6685820

[RO-Crate 1.0] Peter Sefton, Eoghan Ó Carragáin, Stian Soiland-Reyes, Oscar Corcho, Daniel Garijo, Raul Palma, Frederik Coppens, Carole Goble, José María Fernández, Kyle Chard, Jose Manuel Gomez-Perez, Michael R Crusoe, Ignacio Eguinoa, Nick Juty, Kristi Holmes, Jason A. Clark, Salvador Capella-Gutierrez, Alasdair J. G. Gray, Stuart Owen, Alan R. Williams, Giacomo Tartari, Finn Bacall, Thomas Thelen (2019):
RO-Crate Metadata Specification 1.0.
ResearchObject.org / Zenodo
https://doi.org/10.5281/zenodo.3541888
https://w3id.org/ro/crate/1.0

[RO-Crate 1.1] Peter Sefton, Eoghan Ó Carragáin, Stian Soiland-Reyes, Oscar Corcho, Daniel Garijo, Raul Palma, Frederik Coppens, Carole Goble, José María Fernández, Kyle Chard, Jose Manuel Gomez-Perez, Michael R Crusoe, Ignacio Eguinoa, Nick Juty, Kristi Holmes, Jason A. Clark, Salvador Capella-Gutierrez, Alasdair J. G. Gray, Stuart Owen, Alan R. Williams, Giacomo Tartari, Finn Bacall, Thomas Thelen, Hervé Ménager, Laura Rodríguez-Navas, Paul Walk, brandon whitehead, Mark Wilkinson, Paul Groth, Erich Bremer, LJ Garcia Castro, Karl Sebby, Alexander Kanitz, Ana Trisovic, Gavin Kennedy, Mark Graves, Jasper Koehorst, Simone Leo (2020):
RO-Crate Metadata Specification 1.1.
ResearchObject.org / Zenodo
https://doi.org/10.5281/zenodo.4031327
https://w3id.org/ro/crate/1.1

[RO-Crate 1.1.3] Peter Sefton, Eoghan Ó Carragáin, Stian Soiland-Reyes, Oscar Corcho, Daniel Garijo, Raul Palma, Frederik Coppens, Carole Goble, José María Fernández, Kyle Chard, Jose Manuel Gomez-Perez, Michael R Crusoe, Ignacio Eguinoa, Nick Juty, Kristi Holmes, Jason A. Clark, Salvador Capella-Gutierrez, Alasdair J. G. Gray, Stuart Owen, Alan R. Williams, Giacomo Tartari, Finn Bacall, Thomas Thelen, Hervé Ménager, Laura Rodríguez-Navas, Paul Walk, brandon whitehead, Mark Wilkinson, Paul Groth, Erich Bremer, LJ Garcia Castro, Karl Sebby, Alexander Kanitz, Ana Trisovic, Gavin Kennedy, Mark Graves, Jasper Koehorst, Simone Leo, Marc Portier, Paul Brack, Milan Ojsteršek, Bert Droesbeke, Chenxu Niu, Kosuke Tanabe, Tomasz Miksa, Marco La Rosa, Cedric Decruw, Andreas Czerniak, Jeremy Jay, Sergio Serra, Ronald Siebes, Shaun de Witt, Shady El Damaty, Douglas Lowe, Xuanqi Li, Sveinung Gundersen, Muhammad Radifar (2023):
RO-Crate Metadata Specification 1.1.3.
ResearchObject.org / Zenodo
https://doi.org/10.5281/zenodo.7867028
https://w3id.org/ro/crate/1.1

[RO-Crate 1.2] Peter Sefton, Eoghan Ó Carragáin, Stian Soiland-Reyes, Oscar Corcho, Daniel Garijo, Raul Palma, Frederik Coppens, Carole Goble, José María Fernández, Kyle Chard, Jose Manuel Gomez-Perez, Michael R Crusoe, Ignacio Eguinoa, Nick Juty, Kristi Holmes, Jason A. Clark, Salvador Capella-Gutierrez, Alasdair J. G. Gray, Stuart Owen, Alan R. Williams, Giacomo Tartari, Finn Bacall, Thomas Thelen, Hervé Ménager, Laura Rodríguez-Navas, Paul Walk, brandon whitehead, Mark Wilkinson, Paul Groth, Erich Bremer, LJ Garcia Castro, Karl Sebby, Alexander Kanitz, Ana Trisovic, Gavin Kennedy, Mark Graves, Jasper Koehorst, Simone Leo, Marc Portier, Paul Brack, Milan Ojsteršek, Bert Droesbeke, Chenxu Niu, Kosuke Tanabe, Tomasz Miksa, Marco La Rosa, Cedric Decruw, Andreas Czerniak, Jeremy Jay, Sergio Serra, Ronald Siebes, Shaun de Witt, Shady El Damaty, Douglas Lowe, Xuanqi Li, Sveinung Gundersen, Muhammad Radifar, Rudolf Wittner, Oliver Woolland, Paul De Geest, Douglas Fils, Florian Wetzels, Raül Sirvent, Abigail Miller, Jake Emerson, Davide Fucci, Bruno P. Kinoshita, Maciek Bąk, Jens Hollunder, Martin Weise (2024):
RO-Crate Metadata Specification 1.2.0.
(in preparation https://www.researchobject.org/ro-crate/1.2-DRAFT/)
https://doi.org/10.5281/zenodo.8255842
https://w3id.org/ro/crate/1.2

[ro-crate-html-js] npm:
ro-crate-html-js
https://www.npmjs.com/package/ro-crate-html-js

[Robinson 2017] Mark Robinson, Stian Soiland-Reyes, Michael R. Crusoe, Carole Goble (2017):
CWL Viewer: The Common Workflow Language viewer.
F1000Research **6**(ISCB Comm J):1075 (poster)
https://doi.org/10.7490/f1000research.1114375.1

[Robinson 2023] Mark Robinson, Stian Soiland-Reyes, Michael R. Crusoe, Bruno P. Kinoshita, Osakpolor Obaseki, Peter Amstutz, Iacopo Colonnelli, etzanis, Snyk bot, Anushka Shukla, Charles Overbeck, Ward Vandewege, Imgbot, Finn Bacall, Jonathan Leitschuh, yichiehc (2023):
common-workflow-language/cwlviewer: v1.4.7.
Common Workflow Language, GitHub/Zenodo
https://github.com/common-workflow-language/cwlviewer
https://doi.org/10.5281/zenodo.7589709

[Rocca-Serra 2023] Philippe Rocca-Serra, Wei Gu, Vassilios Ioannidis, Tooba Abbassi-Daloii, Salvador Capella-Gutierrez, Ishwar Chandramouliswaran, Andrea Splendiani, Tony Burdett, Robert T. Giessmann, David Henderson, Dominique Batista, Ibrahim Emam, Yojana Gadiya, Lucas Giovanni, Egon Willighagen, Chris Evelo, Alasdair J. G. Gray, Philip Gribbon, Nick Juty, Danielle Welter, Karsten Quast, Paul Peeters, Tom Plasterer, Colin Wood, Eelke van der Horst, Dorothy Reilly, Herman van Vlijmen, Serena Scollen, Allyson Lister, Milo Thurston, Ramon Granell, Gabriel Backianathan, Sebastian Baier, Anne Cambon Thomsen, Martin Cook, Melanie Courtot, Mike d'Arcy, Kurt Dauth, Eva Marin del Piico, Leyla Garcia, Ulrich Goldmann, Valentin Grouès, Daniel J. B. Clarke, Erwan Lefloch, Isuru Liyanage, Petros Papadopoulos, Cyril Pommier, Emiliano Reynares, Francesco Ronzano, Alejandra Delfin-Rossaro, Venkata Sagatopam, Ashni Sedani, Vitaly Sedlyarov, Liubov Shilova, Sukhi Singh, Jolanda Strubel, Kees van Bochove, Zachary Warnes, Peter Woollard, Fuqi Xu, Andrea Zaliani, Susanna-Assunta Sansonem, the FAIR Cookbook Contributors (2023):
The FAIR cookbook - the essential resource for and by FAIR doers.
Scientific Data **10**(10)
https://doi.org/10.1038/s41597-023-02166-3

[Saltz 2006] Joel Saltz, Scott Oster, Shannon Hastings, Stephen Langella, Tahsin Kurc, William Sanchez, Manav Kher, Arumani Manisundaram, Krishnakant Shanbhag, Peter Covitz (2006):
caGrid: design and implementation of the core architecture of the cancer biomedical informatics grid.
Bioinformatics **22**(15)
https://doi.org/10.1093/bioinformatics/btl272

[Samaniego 2010] Luis Samaniego, Rohini Kumar, Sabine Attinger (2010):
Multiscale parameter regionalization of a grid-based hydrologic model at the mesoscale.
https://doi.org/10.1029/2008WR007327

[Samuel 2022] Sheeba Samuel, Birgitta König-Ries (2022):
End-to-End provenance representation for the understandability and reproducibility of scientific experiments using a semantic approach.
Journal of Biomedical Semantics **13**:1
https://doi.org/10.1186/s13326-021-00253-1

[Sandve 2013] Geir Kjetil Sandve, Anton Nekrutenko, James Taylor, Eivind Hovig (2013):
Ten simple rules for reproducible computational research.
PLOS Computational Biology **9**(10):e1003285
https://doi.org/10.1371/journal.pcbi.1003285

[Sandvine 2022] Sandvine (2022):
Global Internet Phenomena Report.
https://www.sandvine.com/global-internet-phenomena-report-2022 (accessed 26 May 2022)

[Sansone 2019] Susanna-Assunta Sansone, Peter McQuilton, Philippe Rocca-Serra, Alejandra Gonzalez-Beltran, Massimiliano Izzo, Allyson L. Lister, Milo Thurston, the FAIRsharing Community (2019):
FAIRsharing as a community approach to standards, repositories and policies.
Nature Biotechnology **37**
https://doi.org/10.1038/s41587-019-0080

[Sauermann 2008] Leo Sauermann, Richard Cyganiak, Danny Ayers, Max Völkel (2008):
Cool URIs for the semantic web.
W3C Interest Group Note
http://www.w3.org/TR/cooluris/

[Scheidegger 2008] Carlos E. Scheidegger, Huy T. Vo, David Koop, Juliana Freire, Claudio T. Silva (2008):
Querying and re-using workflows with VsTrails.
Proceedings of the 2008 ACM SIGMOD international conference on Management of data (*SIGMOD '08*)
https://doi.org/10.1145/1376616.1376747

[schema.org] **Schema.org**
https://schema.org/ (accessed 2021-08-10).

[schema actions] **Schema.org Actions**.
schema.org
https://schema.org/docs/actions.html (accessed 26 May 2022)

[Schouppe 2018] Michel Schouppe, Jean-Claude Burgelman (2018):
Relevance of EOSC and FAIR in the Realm of Open Science and Phases of Implementing the EOSC.
Data Analytics and Management in Data Intensive Domains (DAMDID/RCDL 2018)
CEUR Workshop Proceedings **2277**
https://ceur-ws.org/Vol-2277/paper01.pdf

[Schreiber 2014] Guus Schreiber, Yves Raimond (2014):
RDF 1.1 Primer.
W3C Note
http://www.w3.org/TR/2014/NOTE-rdf11-primer-20140624

[Schriml 2020] Lynn M. Schriml, Maria Chuvochina, Neil Davies, Emiley A. Eloe-Fadrosh, Robert D. Finn, Philip Hugenholtz, Christopher I. Hunter, Bonnie L. Hurwitz, Nikos C. Kyrpides, Folker Meyer, Ilene Karsch Mizrachi, Susanna-Assunta Sansone, Granger Sutton, Scott Tighe, Ramona Walls (2020):
COVID-19 pandemic reveals the peril of ignoring metadata standards.
Scientific Data **7**(1):188
https://doi.org/10.1038/s41597-020-0524-5

[Schröder 2022] Max Schröder, Susanne Staehlke, Paul Groth, J. Barbara Nebe, Sascha Spors, Frank Krüger (2022):
Structure-based knowledge acquisition from electronic lab notebooks for research data provenance documentation.
Journal of Biomedical Semantics **13**:4
https://doi.org/10.1186/s13326-021-00257-x

[Schultes 2019] Erik Schultes, Peter Wittenburg (2019):
FAIR principles and digital objects: Accelerating convergence on a data infrastructure.
Data analytics and management in data intensive domains: 20th international conference (DAMDID/RCDL 2018)
Data analytics and management in data intensive domains: 20th international conference, DAMDID/RCDL 2018, Moscow, Russia, 2018-10-09/–12.
https://doi.org/10.1007/978-3-030-23584-0_1
Preprint: https://doi.org/10.23728/B2SHARE.166A074BFF614A31B05E9DF5BFD9809D

[Schultes 2020] Erik Schultes, Barbara Magagna, Kristina Maria Hettne, Robert Pergl, Marek Suchánek, Tobias Kuhn (2020):
Reusable FAIR implementation profiles as accelerators of FAIR convergence.
International Conference on Conceptual Modeling, ER 2020: Advances in Conceptual Modeling, 2022-11-03/–06, Vienna, Austria.
Lecture notes in Computer Science **12584**
https://doi.org/10.1007/978-3-030-65847-2_13
Preprint: https://doi.org/10.31219/osf.io/2p85g

[Schultes 2022] Erik Schultes, Marco Roos, Luiz Olavo Bonino da Silva Santos, Giancarlo Guizzardi, Jildau Bouwman, Thomas Hankemeier, Arie Baak, Barend Mons (2022):
FAIR Digital Twins for Data-Intensive Research.
Frontiers in Big Data **5**
https://doi.org/10.3389/data.2022.883341

[Schwardmann 2022a] Ulrich Schwardmann, George Strawn, Robert Quick, Peter Wittenburg (2022):
DOIP endorsement request.
FDO Specification Documents PED-DOIPEndorsement-1.1-20221017
FAIR Digital Objects Forum
https://doi.org/10.5281/zenodo.7824796

[Schwardmann 2022b] Ulrich Schwardmann, Tibor Kálmán (2022):
Two Examples on How FDO Types can Support Machine and Human Readability.
Research Ideas and Outcomes **8**:e96014
https://doi.org/10.3897/rio.8.e96014

[Scrocca 2021] Mario Scrocca, Damiano Scandolari, Gloria Re Calegari, Ilaria Baroni, Irene Celino (2021):
The Survey Ontology: Packaging Survey Research as Research Objects.
2nd Workshop on Data and Research Objects Management for Linked Open Science
https://doi.org/10.4126/frl01-006429412

[Sefton 2018] Peter Sefton, Gerard Devine, Christian Evenhuis, Michael Lynch, Sharyn Wise, Michael Lake, Duncan Loxton (2018):
DataCrate: a method of packaging, distributing, displaying and archiving Research Objects.
Workshop on Research Objects (*RO 2018*), 29 Oct 2018 at IEEE eScience 2018, Amsterdam, Netherland.
Zenodo
https://doi.org/10.5281/zenodo.1445817

[Sefton 2021a] Peter Sefton (2021):
FAIR Data Management; It's a lifestyle not a lifecycle.
ptsefton.com.
http://ptsefton.com/2021/04/07/rdmpic/

[Sefton 2021b] Peter Sefton, Mike Lynch, Stian Soiland-Reyes (2021):
GitHub – UTS-eResearch/ro-crate-js: Research Object Crate (RO-Crate) utilities.
https://github.com/UTS-eResearch/ro-crate-js

[Semmler 2022] Tido Semmler, Sergey Danilov, Thomas Rackow, Dmitry Sidorenko, Dirk Barbi, Jan Hegewald, Dmitri Sein, Qiang Wang, Thomas Jung (2022):
IPCC DDC: AWI AWI-CM1.1MR model output prepared for CMIP6 CMIP historical.
https://www.wdc-climate.de/ui/entry?acronym=C6CMAWAWMhi
https://hdl.handle.net/21.14100/2fcf49d3-0608-3373-a47f-0e721b7eaa87

[Serwadda 2018] David Serwadda, Paul Ndebele, M. Kate Grabowski, Francis Bajunirwe, Rhoda K. Wanyenze (2018)
Open data sharing and the Global South—Who benefits?
Science **359**(6376)
https://doi.org/10.1126/science.aap8395

[Shanahan 2021] Hugh Shanahan, Nancy Hoebelheinrich, Angus Whyte (2021):
Progress toward a comprehensive teaching approach to the FAIR data principles.
Patterns **2**(10)
https://doi.org/10.1016/j.patter.2021.100324

[Singhal 2012] Amit Singhal (2012):
Introducing the knowledge graph: Things, not strings.
https://blog.google/products/search/introducing-knowledge-graph-things-not/ (accessed 18 May 2023)

[Sirvent 2022] Raul Sirvent, Javier Conejero, Francesc Lordan, Jorge Ejarque, Laura Rodriguez-Navas, Jose M. Fernandez, Salvador Capella-Gutierrez, Rosa M. Badia (2022):
Automatic, Efficient, and Scalable Provenance Registration for FAIR HPC Workflows.
2022 IEEE/ACM Workshop on Workflows in Support of Large-Scale Science (*WORKS*)
https://doi.org/10.1109/works56498.2022.00006
https://hdl.handle.net/2117/384589

[Śledź 2018] Paweł Śledź, Amedeo Caflisch (2018):
Protein structure-based drug design: from docking to molecular dynamics.
Current Opinion in Structural Biology **48**
https://doi.org/10.1016/j.sbi.2017.10.010

[Smith 2007] Barry Smith, Michael Ashburner, Cornelius Rosse, Jonathan Bard, William Bug, Werner Ceusters, Louis J. Goldberg, Karen Eilbeck, Amelia Ireland, Christopher J. Mungall, Neocles Leontis, Philippe Rocca-Serra, Alan Ruttenberg, Susanna-Assunta Sansone, Richard H. Scheuermann, Nigam Shah, Patricia L. Whetzel, Suzanna Lewis (2007):
The OBO Foundry: Coordinated evolution of ontologies to support biomedical data integration.
Nature Biotechnology **25**(11)
https://doi.org/10.1038/nbt1346

[Smith 2016] Arfon M. Smith, Daniel S. Katz, Kyle E. Niemeyer, FORCE11 Software Citation Working Group (2016):
Software citation principles.
PeerJ Computer Science **2**:e86
https://doi.org/10.7717/peerj-cs.86

[Smith 2022] River Tae Smith, Louisa Willoughby, Trevor Johnston (2022):
Integrating Auslan Resources into the Language Data Commons of Australia.
Proceedings of the LREC2022 10th Workshop on the Representation and Processing of Sign Languages: Multilingual Sign Language Resources.
https://aclanthology.org/2022.signlang-1.28

[Snowley 2023] Kay Snowley, Lara Edwards, Ben Crosby, Helen Tatlow (2023):
Integrating Our Community. Year 1
Health Data Research UK (report)
https://www.hdruk.ac.uk/wp-content/uploads/2023/10/Integrating-Our-Community_v1-Oct-2023-compressed.pdf (accessed 2023-12-06)

[Soiland-Reyes 2014] Stian Soiland-Reyes, Matthew Gamble, Robert Haines (2014):
Research Object Bundle 1.0.
https://w3id.org/bundle/2014-11-05/
https://doi.org/10.5281/zenodo.12586

[Soiland-Reyes 2016] Stian Soiland-Reyes, Pinar Alper, Carole Goble (2016):
Tracking Workflow Execution With TavernaPROV, *ProvenanceWeek 2016*, session “PROV: Three Years Later”
https://s11.no/2016/provweek-tavernaprov/
https://doi.org/10.5281/zenodo.51314

[Soiland-Reyes 2018] Stian Soiland-Reyes, Farah Zaib Khan, Michael R Crusoe (2018):
common-workflow-language/cwlprov: CWLProv 0.6.0.
Zenodo
https://doi.org/10.5281/zenodo.1471585

[Soiland-Reyes 2020a] Stian Soiland-Reyes (2020):
I am looking for which bioinformatics journals encourage authors to submit their code/pipeline/workflow supporting data analysis, *Twitter*
https://twitter.com/soilandreyes/status/1250721245622079488 (archived 2022-12-25 https://web.archive.org/web/20221225100843/https://twitter.com/soilandreyes/status/1250721245622079488)

[Soiland-Reyes 2021] Stian Soiland-Reyes (2021):
Describing and packaging workflows using RO-Crate and BioCompute Objects.
Webinar for U.S. Food and Drug Administration (FDA), 2021-05-12
Zenodo
https://doi.org/10.5281/zenodo.4633732

[Soiland-Reyes 2022a] 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 (2022):
Packaging research artefacts with RO-Crate.
Data Science **5**(2)
https://doi.org/10.3233/DS-210053
https://s11.no/2022/phd/ro-crate/ (Sections 4.1 on page 77 and 4.3 on page 113)

[Soiland-Reyes 2022b] Stian Soiland-Reyes, Genís Bayarri, Pau Andrio, Robin Long, Douglas Lowe, Ania Niewielska, Adam Hospital, Paul Groth (2022):
Making Canonical Workflow Building Blocks interoperable across workflow languages.
Data Intelligence **4**(2)
https://doi.org/10.1162/dint_a_00135
https://s11.no/2022/phd/canonical-workflow-building-blocks/ (Section 5.1 on page 121)

[Soiland-Reyes 2022c] Stian Soiland-Reyes, Peter Sefton, Leyla Jael Castro, Frederik Coppens, Daniel Garijo, Simone Leo, Marc Portier, Paul Groth (2022):
Creating lightweight FAIR digital objects with RO-Crate.
Research Ideas and Outcomes **8**:e93937
https://doi.org/10.3897/rio.8.e93937
https://s11.no/2022/phd/fdo-with-ro-crate/ (Section 4.2 on page 109)

[Soiland-Reyes 2022d] Stian Soiland-Reyes, Leyla Jael Castro, Daniel Garijo, Marc Portier, Carole Goble, Paul Groth (2022):
Updating Linked Data practices for FAIR Digital Object principles.
Research Ideas and Outcomes **8**:e94501
https://doi.org/10.3897/rio.8.e94501
https://s11.no/2022/phd/updating-ld-for-fdo/ (Section 3.2 on page 71)

[Soiland-Reyes 2022e] Stian Soiland-Reyes,Bruno P. Kinoshita (2022):
stain/signposting: Signposting v0.9.0.
Zenodo
https://doi.org/10.5281/zenodo.7256713

[Soiland-Reyes 2022f] Stian Soiland-Reyes, Carole Goble (2022):
EuroScienceGateway: WP2 introduction.
EuroScienceGateway kickoff 2022-10-06
https://doi.org/10.5281/zenodo.7152762

[Soiland-Reyes 2022g] 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 (2022):
Packaging research artefacts with RO-Crate.
RO-Crate
Zenodo
https://w3id.org/ro/doi/10.5281/zenodo.5146227
https://doi.org/10.5281/zenodo.5833456

[Soiland-Reyes 2023a] Stian Soiland-Reyes (2023):
Comparison tables for evaluating FAIR Digital Object and Linked Data.
RO-Crate.
Zenodo
https://w3id.org/ro/doi/10.5281/zenodo.8075229
https://doi.org/10.5281/zenodo.8075229

[Soiland-Reyes 2023b] Stian Soiland-Reyes, Herbert Van de Sompel (2023):
Enabling FAIR Signposting and RO-Crate for content/metadata discovery and consumption.
Webinar: FAIR-IMPACT Open Call for Support, 2023-03-27
Zenodo
https://doi.org/10.5281/zenodo.7774582

[Soiland-Reyes 2023c] Stian Soiland-Reyes, Carole Goble (2023):
Building diverse FDO Collections using RO-Crate.
FAIR Digital Object Forum, workshop ”Defining FDO Collections”, 2023-04-14.
https://doi.org/10.5281/zenodo.7828632

[Soiland-Reyes 2023d] Stian Soiland-Reyes, Stuart Wheater (2023):
Five Safes RO-Crate profile, version 0.4.
TRE-FX Candidate Recommendation
https://w3id.org/5s-crate/0.4

[Soiland-Reyes 2023e] Stian Soiland-Reyes, Stuart Wheater, Thomas Giles, Carole Goble, Philip Quinlan (2023):
TRE-FX Technical Documentation - Five Safes RO-crate.
Zenodo
https://doi.org/10.5281/zenodo.10376350

[Soiland-Reyes 2024a] Stian Soiland-Reyes, Leyla Jael Castro, Rohitha Ravinder, Claus Weiland, Jonas Grieb, Alexander Rogers, Christophe Blanchi, Herbert Van de Sompel (2024):
BioHackEU23 report: **Enabling FAIR Digital Objects with RO-Crate, Signposting and Bioschemas**.
BioHackrXiv
https://doi.org/10.37044/osf.io/gmk2h
https://s11.no/2024/enabling-fair-digital-objects/ (Supplement 20)

[Soiland-Reyes 2024b] Stian Soiland-Reyes, Carole Goble, Paul Groth (2024):
Evaluating FAIR Digital Object and Linked Data as distributed object systems.
PeerJ Computer Science **10**:e1781
arXiv 2306.07436
https://doi.org/10.7717/peerj-cs.1781
https://s11.no/2023/phd/evaluating-fdo/ (Sections 3.2.1 on page 71 and 3.1 on page 31)

[Soiland-Reyes 2024c] Stian Soiland-Reyes, Peter Sefton, Simone Leo, Leyla Jael Castro, Claus Weiland, Herbert Van de Sompel (2024):
Practical webby FDOs with RO-Crate and FAIR Signposting: Experiences and lessons learned.
International FAIR Digital Objects Implementation Summit 2024, Berlin, Germany, 2024-03-20/–21.
Open Conference Proceedings **4** (submitted)
https://s11.no/2024/webby-fdos/ (Supplement 22)

[Speicher 2015] Steve Speicher, John Arwe, Ashok Malhotra (eds) (2015):
Linked Data Platform 1.0.
W3C Recommendation 26 February 2015
http://www.w3.org/TR/2015/REC-ldp-20150226/

[Sporny 2014] Manu Sporny, Dave Longley, Gregg Kellogg, Markus Lanthaler Niklas Lindström (2014):
JSON-LD 1.0: A JSON-based Serialization for Linked Data.
W3C Recommendation 16 January 2014
https://www.w3.org/TR/2014/REC-json-ld-20140116/

[Sporny 2015] Manu Sporny, Ivan Herman, Ben Adida, Mark Birbeck (2015):
RDFa 1.1 Primer - Third Edition.
W3C Working Group Note 17 March 2015
https://www.w3.org/TR/2015/NOTE-rdfa-primer-20150317/

[Sporny 2020] Manu Sporny, Dave Longley, Gregg Kellogg, Markus Lanthaler, Pierre-Antoine Champin, Niklas Lindström (2020):
JSON-LD 1.1: A JSON-based Serialization for Linked Data.
W3C Recommendation 16 July 2020
https://www.w3.org/TR/2020/REC-json-ld11-20200716/

[Sporny 2023] Manu Sporny, Amy Guy (2023):
Media Types with Multiple Suffixes.
Internet Engineering Task Force
https://datatracker.ietf.org/doc/draft-ietf-mediaman-suffixes/03/

[Stallings 1990] William Stallings (1990):
Handbook of computer-communications standards: The open systems (OSI) model and OSI-related standards, 2nd ed.
Sams.
ISBN 978-0-672-22697-7

[Stanczyk 1987] Stefan K. Stanczyk (1987):
Process modelling for information system description.
The Open University
https://doi.org/10.21954/ou.ro.0000f821

[Stefi 2015a] Anisa Stefi, Thomas Hess (2015):
To develop or to reuse? Two perspectives on external reuse in software projects.
International Conference of Software Business (ICSOB 2015), Braga, Portugal, 2015-06-10/–12.
ICSOB 2015: Software business
http://doi.org/10.1007/978-3-319-19593-3_18

[Stefi 2015b] Anisa Stefi (2015):
Do Developers Make Unbiased Decisions? - The Effect of Mindfulness and Not-Invented-Here Bias on the Adoption of Software Components.
European Conference on Information Systems (ECIS 2015), Münster, Germany, 2015-05-26/–29
https://doi.org/10.18151/7217489

[Stodden 2016] Victoria Stodden, Marcia McNutt, David H. Bailey, Ewa Deelman, Yolanda Gil, Brooks Hanson, Michael A. Heroux, John P.A. Ioannidis, Michela Taufer (2016):
Enhancing reproducibility for computational methods.
Science **354**(6317)
https://doi.org/10.1126/science.aah6168

[Suetake 2022] Hirotaka Suetake, Tomoya Tanjo, Manabu Ishii, Bruno P. Kinoshita, Takeshi Fujino, Tsuyoshi Hachiya, Yuichi Kodama, Takatomo Fujisawa, Osamu Ogasawara, Atsushi Shimizu, Masanori Arita, Tsukasa Fukusato, Takeo Igarashi, Tazro Ohta (2022):
Sapporo: A workflow execution service that encourages the reuse of workflows in various languages in bioinformatics [version 1; peer review: 2 approved with reservations].
F1000Research **11**:889
https://doi.org/10.12688/f1000research.122924.1

[Suetake 2023a] Hirotaka Suetake, Tsukasa Fukusato, Takeo Igarashi, Tazro Ohta (2023):
A workflow reproducibility scale for automatic validation of biological interpretation results.
GigaScience **12**:iad031
https://doi.org/10.1093/gigascience/giad031

[Suetake 2023b] Hirotaka Suetake, Tazro Inutano Ohta, Tomoya Tanjo, Manabu ISHII, Bruno P. Kinoshita, DrYak (2023):
sapporo-wes/sapporo-service: 1.5.1
Zenodo
https://doi.org/10.5281/zenodo.10134452

[Sun 2003a] Sam Sun, Larry Lannom, Brian P. Boesch (2003):
Handle System Overview.
RFC Editor, RFC 3650
https://doi.org/10.17487/rfc3650

[Sun 2003b] Sam Sun, Sean Reilly, Larry Lannom, Jason Petrone (2003):
Handle System Protocol (ver 2.1) Specification.
RFC Editor, RFC 3652
https://doi.org/10.17487/rfc3652

[Sweeney 2018] Patrick W. Sweeney, Binil Starly, Paul J. Morris, Yiming Xu, Aimee Jones, Sridhar Radhakrishnan, Christopher J. Grassa, Charles C. Davis (2018):
Large-scale digitization of herbarium specimens: Development and usage of an automated, high-throughput conveyor system.
Taxon **67**(1)
https://doi.org/10.12705/671.10

[Taschuk 2017] Morgan Taschuk, Greg Wilson (2017):
Ten simple rules for making research software more robust.
PLOS Computational Biology **13**(4):e1005412
https://doi.org/10.1371/journal.pcbi.1005412

[Taivalsaari 2021] Antero Taivalsaari, Tommi Mikkonen, Cesare Pautasso, Kari Systä (2021):
Full Stack Is Not What It Used to Be. *Web Engineering* (ICWE 2021)
Lecture Notes in Computer Science **12706**
https://doi.org/10.1007/978-3-030-74296-6_28

[Tegelberg 2017] Riitta Tegelberg, Jere Kahanpaa, Janne Karppinen, Tero Mononen, Zhenzhe Wu, Hannu Saarenmaa (2017):
Mass Digitization of Individual Pinned Insects Using Conveyor-Driven Imaging.
2017 IEEE 13th International Conference on E-Science (e-Science)
https://doi.org/10.1109/eScience.2017.85

[Tejedor 2017] Enric Tejedor, Yolanda Becerra, Guillem Alomar, Anna Queralt, Rosa M. Badia, Jordi Torres, Toni Cortes, Jesús Labarta (2017):
PyCOMPSs: Parallel computational workflows in Python.
The International Journal of High Performance Computing Applications **31**(1)
https://doi.org/10.1177/1094342015594678

[Thieberger 2012] N. Thieberger, L. Barwick (2012):
Keeping records of language diversity in melanesia: The Pacific and regional archive for digital sources in endangered cultures (PARADISEC).
Melanesian Languages on the Edge of Asia: Challenges for the 21st Century, N. Evans and M. Klamer (eds.)
Language Documentation & Conservation Special Publication **SP05**
https://hdl.handle.net/10125/4567

[Thiers 2016] Barbara M. Thiers, Melissa C. Tulig, Kimberly A. Watson (2016):
Digitization of the New York Botanical Garden herbarium.
Brittonia **68**(3)
https://doi.org/10.1007/s12228-016-9423-7

[Thompson 2012] Henry Thompson, Sandy Gao, David Beech, Murray Maloney, Noah Mendelsohn, Michael Sperberg-McQueen (2012):
W3C XML Schema Definition Language (XSD) 1.1 Part 1: Structures.
W3C Recommendation 5 April 2012
https://www.w3.org/TR/2012/REC-xmlschema11-1-20120405/

[Thompson 2020] Mark Thompson, Kees Burger, Rajaram Kaliyaperumal, Marco Roos, Luiz Olavo Bonino da Silva Santos (2020):
Making FAIR Easy with FAIR Tools: From Creolization to Convergence.
Data Intelligence **2**(1-2)
https://doi.org/10.1162/dint_a_00031

[Thornton 2019] Katherine Thornton, Harold Solbrig, Gregory S. Stupp, Jose Emilio Labra Gayo, Daniel Mietchen, Eric Prud, Andra Waagmeester (2019):
Using shape expressions (ShEx) to share RDF data models and to guide curation with rigorous validation.
The Semantic Web: 16th international conference, (ESWC 2019), Portorož, Slovenia, 2019-06-02/–06
https://doi.org/10.1007/978-3-030-21348-0_39

[Tirmizi 2011] Syed Tirmizi, Stuart Aitken, Dilvan A. Moreira, Chris Mungall, Juan Sequeda, Nigam H. Shah, Daniel P. Miranker (2011):
Mapping between the OBO and OWL ontology languages.
Journal of Biomedical Semantics **2**:S3
https://doi.org/10.1186/2041-1480-2-s1-s3

[Tran 2014] **Linked Data Mashups: A Review on Technologies, Applications and Challenges**.
Intelligent Information and Database Systems. ACIIDS 2014. Ngoc Thanh Nguyen, Boonwat Attachoo, Bogdan Trawiński, Kulwadee Somboonviwat (eds.)
Lecture Notes in Computer Science **8398**
https://doi.org/10.1007/978-3-319-05458-2_27

[Trautwein 2022] Dennis Trautwein, Aravindh Raman, Gareth Tyson, Ignacio Castro, Will Scott, Moritz Schubotz, Bela Gipp, Yiannis Psaras (2022):
Design and Evaluation of IPFS: A Storage Layer for the Decentralized Web.
Proceedings of the ACM SIGCOMM 2022 Conference
arXiv:2208.05877
https://doi.org/10.1145/3544216.3544232

[Triki 2020] Abdelaziz Triki, Bassem Bouaziz, Walid Mahdi, Jitendra Gaikwad (2020):
Objects Detection from Digitized Herbarium Specimen based on Improved YOLO V3.
Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications **4**
https://doi.org/10.5220/0009170005230529

[Troncy 2010] Raphaël Troncy, Werner Bailer, Martin Höffernig, Michael Hausenblas (2010):
VAMP: A service for validating MPEG-7 descriptions w.r.t. to formal profile definitions.
Multimedia tools and applications **46**(2–3)
https://www.persistent-identifier.nl/urn:nbn:nl:ui:18-14511
https://doi.org/10.1007/s11042-009-0397-2

[Tudorache 2020] Tania Tudorache (2020):
Ontology engineering: Current state, challenges, and future directions.
Semantic Web **11**(1)
https://doi.org/10.3233/SW-190382

[Tupelo-Scheck 2022] Robert Tupelo-Schneck, Larry Lannom (2022):
Brief Introduction to Cordra & DOIP.
RDA FAIR DO Fabric, 2022-03-22
https://www.rd-alliance.org/sites/default/files/Cordra.2022.pdf

[Turcoane 2014] Ovidiu Turcoane (2014):
Linked data, JSON-LD and the semantics of cultural and scientific heritage.
Digital Presentation and Preservation of Cultural and Scientific Heritage **4**
https://doi.org/10.55630/dipp.2014.4.11

[Unger 2016] Jakob Unger, Dorit Merhof, Susanne Renner (2016):
Computer vision applied to herbarium specimens of German trees: testing the future utility of the millions of herbarium specimen images for automated identification
. *BMC Evolutionary Biology* **16**(1)
https://doi.org/10.1186/s12862-016-0827-5

[Van de Sompel 2007] Herbert Van de Sompel, Carl Lagoze (2007):
Interoperability for the discovery, use, and re-use of units of scholarly communication.
CTWatch Quarterly **3**(3)
http://icl.utk.edu/ctwatch/quarterly/articles/2007/08/interoperability-for-the-discovery-use-and-re-use-of-units-of-scholarly-communication/

[Van de Sompel 2013] Herbert Van de Sompel, Michael Nelson, Robert Sanderson (2013):
HTTP Framework for Time-Based Access to Resource States –Memento.
RFC Editor, RFC 7089
https://doi.org/10.17487/rfc7089

[Van de Sompel 2015] Herbert Van de Sompel, Michael L. Nelson (2015):
Reminiscing About 15 Years of Interoperability Efforts.
D-Lib Magazine **21**(11/12)
https://doi.org/10.1045/november2015-vandesompel

[Van de Sompel 2022] Herbert Van de Sompel, Martin Klein, Shawn Jones, Michael L. Nelson, Simeon Warner, Anusuriya Devaraju, Robert Huber, Wilko Steinhoff, Vyacheslav Tykhonov, Luc Boruta, Enno Meijers, Stian Soiland-Reyes, Mark Wilkinson (2022):
FAIR Signposting Profile. (version 20220727)
https://signposting.org/FAIR/

[Van de Sompel 2023] Herbert van de Sompel (2023):
FAIR Digital Objects and FAIR Signposting.
FAIR Digital Object forum webinar.
https://doi.org/10.5281/zenodo.7977333

[Verborgh 2018] Ruben Verborgh (2018):
Designing a Linked Data developer experience.
https://ruben.verborgh.org/blog/2018/12/28/designing-a-linked-data-developer-experience/
(accessed 26 May 2022)

[Verborgh 2020] Ruben Verborgh, Miel Vander Sande (2020):
The semantic web identity crisis: In search of the trivialities that never were.
Semantic Web **11**(1)
https://doi.org/10.3233/SW-190372

[Verburg 2023] Maaike Verburg, Robert Huber, Clement Jonquet, Daniel Garijo (2023):
FAIR-IMPACT project response to ”FAIR Assessment Tools: Towards an ”Apples to Apples” Comparisons”.
Zenodo
https://doi.org/10.5281/zenodo.7848102

[Vergoulis 2021] Thanasis Vergoulis (2021): **Use of RO-Crates in SCHeMa**.
RO-Crate community call, 2021-04-08
https://doi.org/10.5281/zenodo.4671709

[Vergoulis 2022] Thanasis Vergoulis, Konstantinos Zagganas, Loukas Kavouras, Martin Reczko, Stelios Sartzetakis, Theodore Dalamagas (2022):
SCHeMa: Scheduling Scientific Containers on a Cluster of Heterogeneous Machines.
SSDBM 2021: 33rd International Conference on Scientific and Statistical Database Management
arXiv:2103.13138
https://doi.org/10.1145/3468791.3468813

[de Visser 2023] Casper de Visser, Lennart F. Johansson, Purva Kulkarni, Hailiang Mei, Pieter Neerincx, K. Joeri van der Velde, Péter Horvatovich, Alain J. van Gool, Morris A. Swertz, Peter A. C. ‘t Hoen, Anna Niehues (2023):
Ten quick tips for building FAIR workflows.
PLOS Computational Biology **19**(9): e1011369
https://doi.org/10.1371/journal.pcbi.1011369

[Vivian 2017] John Vivian, Arjun Arkal Rao, Frank Austin Nothaft, Christopher Ketchum, Joel Armstrong, Adam Novak, Jacob Pfeil, Jake Narkizian, Alden D Deran, Audrey Musselman-Brown, Hannes Schmidt, Peter Amstutz, Brian Craft, Mary Goldman, Kate Rosenbloom, Melissa Cline, Brian O’Connor, Megan Hanna, Chet Birger, W James Kent, David A Patterson, Anthony D Joseph, Jingchun Zhu, Sasha Zaranek, Gad Getz, David Haussler, Benedict Paten (2017):
Toil enables reproducible, open source, big biomedical data analyses.
Nature Biotechnology **35**(4)
https://doi.org/10.1038/nbt.3772

[Volk 2014] Carol J. Volk, Yasmin Lucero, Katie Barnas (2014):
Why is data sharing in collaborative natural resource efforts so hard and what can we do to improve it?
Environmental Management **x53**(5)
https://doi.org/10.1007/s00267-014-0258-2

[W3C 2007] W3C Technical Architecture Group (2007):
Dereferencing HTTP URIs.
Draft Tag Finding.
https://www.w3.org/2001/tag/doc/httpRange-14/2007-08-31/HttpRange-14.html

[W3C 2012] W3C OWL Working Group (2012):
OWL 2 Web Ontology Language Document Overview (Second Edition).
W3C Recommendation 11 December 2012
https://www.w3.org/TR/2012/REC-owl2-overview-20121211

[W3C 2013] The W3C SPARQL Working Group (2013):
SPARQL 1.1 Overview.
W3C Recommendation 21 March 2013
https://www.w3.org/TR/2013/REC-sparql11-overview-20130321/ (accessed 26 May 2022)

[W3C 2015] W3C (2015):
Linked Data.
https://www.w3.org/standards/semanticweb/data (accessed 26 May 2022)

[W3Techs 2023] W3Techs 2023:
Usage Statistics of JSON-LD for Websites. 2023-05.
W3Techs - World Wide Web Technology Surveys, Q-Success.
https://w3techs.com/technologies/details/da-jsonld (accessed 18 May 2023)

[Walton 2020a] Stephanie Walton, Laurence Livermore, Olaf Bánki, Robert Cubey, Robyn Drinkwater, Markus Englund, Carole Goble, Quentin Groom, Christopher Kermorvant, Isabel Rey, Celia Santos, Ben Scott, Alan Williams, Zhengzhe Wu (2020):
Landscape Analysis for the Specimen Data Refinery.
Research Ideas and Outcomes **6**:e57602
https://doi.org/10.3897/rio.6.e57602

[Walton 2020b] Stephanie Walton, Laurence Livermore, Mathias Dillen, Sofie De Smedt, Quentin Groom, Anne Koivunen, Sarah Phillips (2020):
A cost analysis of transcription systems.
Research Ideas and Outcomes **6**:e56211.
https://doi.org/10.3897/rio.6.e56211

[Watanabe 2019] Myrna E Watanabe (2019):
The Evolution of Natural History Collections: New research tools move specimens, data to center stage.
BioScience **69**(3)
https://doi.org/10.1093/biosci/biy163

[WHATWG 2023] WHATWG (2023):
Microdata.
HTML Living Standard
https://html.spec.whatwg.org/multipage/microdata.html (accessed 13 June 2023)

[Weigel 2018] Tobias Weigel, Beth Plale, Mark Parsons, Gabriel Zhou, Yu Luo, Ulrich Schwardmann, Robert Quick, Margareta Hellström, Kei Kurakawa (2018):
RDA Recommendation on PID Kernel Information
Research Data Alliance
https://doi.org/10.15497/rda00031

[Weigel 2022] Daan Broeder, Peter Wittenburg, Ivonne Anders, Karsten Peters-von Gehlen (2022):
FDO – kernel attributes & metadata.
FDO Specification Documents PR-FDO-KernelAttributesAndMetadata-2.0-20221017
FAIR Digital Objects Forum
https://doi.org/10.5281/zenodo.7825693

[Weiland 2022a] Claus Weiland, Ulrich Schwardmann, Peter Wittenburg, Christine Kirkpatrick, Robert Hanisch, Zachary Trautt (2022):
FAIR Digital Objects Forum Document Standards WD-DocProcessStd-1.1-20220129 (internal draft)
FAIR Digital Objects Forum
https://zenodo.org/doi/10.5281/zenodo.10943371

[Weiland 2022b] Claus Weiland, Sharif Islam, Daan Broder, Ivonne Anders, Peter Wittenburg (2022):
FDO machine actionability. Version 2.2
FAIR Digital Objects Forum Document Standards PR-MachineActionDef-2.2-20221119
FAIR Digital Objects Forum
https://doi.org/10.5281/zenodo.7825650

[Whetzel 2011] Patricia L Whetzel, Natalya F Noy, Nigam H Shah, Paul R Alexander, Csongor Nyulas, Tania Tudorache, Mark A Musen (2011):
BioPortal: enhanced functionality via new Web services from the National Center for Biomedical Ontology to access and use ontologies in software applications.
Nucleic Acids Research **39**(Web Server issue):W541-5
https://doi.org/10.1093/nar/gkr469

[de Wit 2022] Renske de Wit (2022):
A Non-Intimidating Approach to Workflow Reproducibility in Bioinformatics: Adding Metadata to Research Objects through the Design and Evaluation of Use-Focused Extensions to CWLProv.
Zenodo
https://doi.org/10.5281/zenodo.7113250

[de Wit 2023] Renske de Wit, Michael R Crusoe (2023):
Analysis of runcrate.
Zenodo
https://doi.org/10.5281/zenodo.10251812

[Wieczorek 2012] John Wieczorek, David Bloom, Robert Guralnick, Stan Blum, Markus Döring, Renato Giovanni, Tim Robertson, David Vieglais (2012):
Darwin Core: An Evolving Community-Developed Biodiversity Data Standard.
PLOS ONE **7**(1):e29715
https://doi.org/10.1371/journal.pone.0029715

[Wilde 2013] Erik Wilde (2013):
The 'profile' Link Relation Type.
RFC Editor, RFC 6906
https://doi.org/10.17487/rfc6906

[Wilde 2020] Erik Wilde, Herbert Van de Sompel (2022):
Linkset: Media Types and a Link Relation Type for Link Sets.
RFC Editor, RFC 9264
https://doi.org/10.17487/rfc9264

[Wilkinson 2016] Mark D. Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E. Bourne, Jildau Bouwman, Anthony J. Brookes, Tim Clark, Mercè Crosas, Ingrid Dillo, Olivier Dumon, Scott Edmunds, Chris T. Evelo, Richard Finkers, Alejandra Gonzalez-Beltran, Alasdair J.G. Gray, Paul Groth, Carole Goble, Jeffrey S. Grethe, Jaap Heringa, Peter A.C 't Hoen, Rob Hooft, Tobias Kuhn, Ruben Kok, Joost Kok, Scott J. Lusher, Maryann E. Martone, Albert Mons, Abel L. Packer, Bengt Persson, Philippe Rocca-Serra, Marco Roos, Rene van Schaik, Susanna-Assunta Sansone, Erik Schultes, Thierry Sengstag, Ted Slater, George Strawn, Morris A. Swertz, Mark Thompson, Johan van der Lei, Erik van Mulligen, Jan Velterop, Andra Waagmeester, Peter Wittenburg, Katherine Wolstencroft, Jun Zhao, Barend Mons (2016):
The FAIR Guiding Principles for scientific data management and stewardship.
Scientific Data **3**(1):160018
https://doi.org/10.1038/sdata.2016.18

[Wilkinson 2018] Mark D. Wilkinson, Susanna-Assunta Sansone, Erik Schultes, Peter Doorn, Luiz Olavo Bonino da Silva Santos, Michel Dumontier (2018):
A design framework and exemplar metrics for FAIRness.
Scientific Data **5**:180118
https://doi.org/10.1038/sdata.2018.118

[Wilkinson 2022a] Mark D. Wilkinson, Susanna-Assunta Sansone, Grootveld Marjan, Josefine Nordling, Richard Dennis, David Hecker (2022):
FAIR Assessment Tools: Towards an "Apples to Apples" Comparisons.
EOSC FAIR Metrics and Data Quality Task Force
EOSC Association / Zenodo
https://doi.org/10.5281/zenodo.7463421

[Wilkinson 2022b] Sean R. Wilkinson, Greg Eisenhauer, Anuj J. Kapadia, Kathryn Knight, Jeremy Logan, Patrick Widener, Matthew Wolf (2022):
F* workflows: When parts of FAIR are missing**.
IEEE 18th International Conference on e-Science (e-Science 2022)
arXiv:2209.09022
https://doi.org/10.1109/eScience55777.2022.00090

[Wilkinson 2023a] Mark D. Wilkinson, Susanna-Assunta Sansone, Eva Méndez, Romain David, Richard Dennis, David Hecker, Mari Kleemola, Carlo Lacagnina, Anastasija Nikiforova, Leyla Jael Castro (2023):
Community-driven governance of FAIRness assessment: an open issue, an open discussion.
Open Research Europe **2**
https://doi.org/10.12688/openreseurope.15364.2

[Wilkinson 2024] Mark D Wilkinson, Susanna-Assunta Sansone, Marjan Grootveld, Richard Dennis, David Hecker, Robert Huber, Stian Soiland-Reyes, Herbert Van de Sompel, Andreas Czerniak, Milo Thurston, Allyson L. Lister, Alban Gaignard (2024):
Report on "FAIR Signposting" and its uptake by the community.
EOSC FAIR Metrics and Data Quality Task Force
https://doi.org/10.5281/zenodo.10490289
https://s11.no/2024/signposting-report/ (Supplement 21)

[Williams 2012] Antony J. Williams, Lee Harland, Paul Groth, Stephen Pettifer, Christine Chichester, Egon L. Willighagen, Chris T. Evelo, Niklas Blomberg, Gerhard Ecker, Carole Goble, Barend Mons (2012):
Open PHACTS: Semantic interoperability for drug discovery.
Drug Discovery Today **17**(21-22)
https://doi.org/10.1016/j.drudis.2012.05.016

[Wittenburg 2019] Peter Wittenburg, George Strawn, Barend Mons, Luiz Bonino, Erik Schultes (2019):
Digital objects as drivers towards convergence in data infrastructures.
B2Share
https://doi.org/10.23728/b2share.b605d85809ca45679b110719b6c6cb11

[Wittenburg 2022a] Peter Wittenburg, Ivonne Anders, Christophe Blanchi, Merret Buurman, Carole Goble, Jonas Grieb, Alex Hardisty, Sharif Islam, Thomas Jejkal, Tibor Kálmán, Christine Kirkpatrick, Laurence Lannom, Thomas Lauer, Giridhar Manepalli, Karsten Peters-von Gehlen, Andreas Pfeil, Robert Quick, Mark Sanden, Ulrich Schwardmann, Stian Soiland-Reyes, Rainer Stotzka, Zachary Trautt, Dieter Van Uytvanck, Claus Weiland, Philipp Wieder (2022):
FAIR digital object demonstrators 2021.
Zenodo
https://doi.org/10.5281/zenodo.5872645

[Wittenburg 2022b] Peter Wittenburg, Alex Hardisty, Yann Le Franc, Amirpasha Mozaffari, Limor Peer, Nikolay A. Skvortsov, Zhiming Zhao, Alessandro Spinuso (2022):
Canonical Workflows to Make Data FAIR.
Data Intelligence **4**(2)
https://doi.org/10.1162/dint_a_00132

[Wittenburg 2023a] Peter Wittenburg, Ulrich Schwardmann, Christophe Blanchi, Claus Weiland (2023):
FDOs to Enable Cross-Silo Work.
Proceedings of the Conference on Research Data Infrastructure **1**
https://doi.org/10.52825/cordi.v1i.263

[Wittenburg 2023b] Peter Wittenburg, Dimitris Koureas (2023):
FDO to Structure the Domain of Knowledge.
Proceedings of the Conference on Research Data Infrastructure **1**
https://doi.org/10.52825/cordi.v1i.374

[Wittner 2020] Rudolf Wittner, Petr Holub, Heimo Müller, Joerg Geiger, Carole Goble, Stian Soiland-Reyes, Luca Pireddu, Francesca Frexia, Cecilia Mascia, Elliot Fairweather, Jason R. Swedlow, Josh Moore, Caterina Strambio, David Grunwald, Hiroki Nakae (2020):
ISO 23494: Biotechnology - Provenance Information Model for Biological Specimen and Data. In: Glavic B., Braganholo V., Koop D. (eds) *Provenance and Annotation of Data and Processes* (IPAW 2020/2021).
Lecture Notes in Computer Science 12839
https://doi.org/10.1007/978-3-030-80960-7_16
https://s11.no/2021/phd/iso-23494-provenance/ (Supplement 7)

[Wittner 2022] Rudolf Wittner, Cecilia Mascia, Matej Gallo, Francesca Frexia, Heimo Müller, Markus Plass, Jörg Geiger, Petr Holub (2022):
Lightweight Distributed Provenance Model for Complex Real–world Environments.
Scientific Data **9**:503
https://doi.org/10.1038/s41597-022-01537-6

[Wittner 2023a] Rudolf Wittner, Petr Holub, Cecilia Mascia, Francesca Frexia, Heimo Müller, Markus Plass, Clare Allocca, Fay Betsou, Tony Burdett, Ibon Cancio, Adriane Chapman, Martin Chapman, Mélanie Courtot, Vasa Curcin, Johann Eder, Mark Elliot, Katrina Exter, Carole Goble, Martin Golebiewski, Bron Kisler, Andreas Kremer, Simone Leo, Sheng Lin-Gibson, Anna Marsano, Marco Mattavelli, Josh Moore, Hiroki Nakae, Isabelle Perseil, Ayat Salman, James Sluka, Stian Soiland-Reyes, Caterina Strambio-De-Castillia, Michael Sussman, Jason R. Swedlow, Kurt Zatloukal, Jörg Geiger (2023):
Toward a common standard for data and specimen provenance in life sciences.
Learning Health Systems :e10365
https://doi.org/10.1002/lrh2.10365

[Wittner 2023b] Rudolf Wittner, Matej Gallo, Simone Leo, Cecilia Mascia, Francesca Frexia, Markus Plass, Stian Soiland-Reyes, Heimo Müller, Jörg Geiger, Petr Holub (2023):
Linking provenance and its metadata in multi-organizational environments of life sciences.
Submitted (PeerJ Computer Science)
https://s11.no/2023/phd/linking-provenance/ (Supplement 17)

[Wittner 2023c] Rudolf Wittner, Matej Rudolf, Simone Leo, Stian Soiland-Reyes (2023):
Packing provenance using CPM RO-Crate profile (Version 1.1)
Data set.
Zenodo
https://doi.org/10.5281/zenodo.8095888

[Wolstencroft 2011] Katy Wolstencroft, Stuart Owen, Matthew Horridge, Olga Krebs, Wolfgang Mueller, Jacky L. Snoep, Franco du Preez, Carole Goble (2011):
RightField: Embedding ontology annotation in spreadsheets.
Bioinformatics **27**(14)
https://doi.org/10.1093/bioinformatics/btr312

[Wolstencroft 2013] Katherine Wolstencroft, Robert Haines, Donal Fellows, Alan Williams, David Withers, Stuart Owen, Stian Soiland-Reyes, Ian Dunlop, Aleksandra Nenadic, Paul Fisher, Jiten Bhagat, Khalid Belhajjame, Finn Bacall, Alex Hardisty, Abraham Nieva de la Hidalga, Maria P. Balcazar Vargas, Shoaib Sufi, Carole Goble (2013):
The Taverna workflow suite: Designing and executing workflows of Web Services on the desktop, web or in the cloud.
Nucleic Acids Research **41**(W1)
https://doi.org/10.1093/nar/gkt328

[Wood 2014] David Wood, Richard Cyganiak, Markus Lanthaler (2014):
RDF 1.1 Concepts and Abstract Syntax.
W3C Recommendation 25 February 2014
https://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/

[Woolland 2022] Oliver Woolland, Paul Brack, Stian Soiland-Reyes, Ben Scott, Laurence Livermore (2022):
Incrementally building FAIR Digital Objects with Specimen Data Refinery workflows.
1st International Conference on FAIR Digital Objects (FDO 2022) (poster)
Research Ideas and Outcomes **8**:e94349
https://doi.org/10.3897/rio.8.e94349
https://s11.no/2022/phd/incrementally-building-fdos/
(see Section 5.3 on page 148)

[WorkflowHub 2023] WorkflowHub (2023):
WorkflowHub project: Project pages for developing and running the WorkflowHub, a registry of scientific workflows.
https://w3id.org/workflowhub/

[Wright 2022] Austin Wright, Henry Andrews, Ben Hutton, Greg Dennis (2022):
JSON schema: A media type for describing JSON documents.
Internet Engineering Task Force
https://datatracker.ietf.org/doc/draft-bhutton-json-schema/01/

[WRROC 2023a] Workflow Run RO-Crate working group (2023):
Process Run Crate specification. Version 0.4
Zenodo
https://w3id.org/ro/wfrun/process/0.4
https://doi.org/10.5281/zenodo.10203944

[WRROC 2023b] Workflow Run RO-Crate working group (2023):
Workflow Run Crate specification. Version 0.4
Zenodo
https://w3id.org/ro/wfrun/workflow/0.4
https://doi.org/10.5281/zenodo.10203971

[WRROC 2023c] Workflow Run RO-Crate working group (2023):
Provenance Run Crate specification. Version 0.4
Zenodo
https://w3id.org/ro/wfrun/provenance/0.4
https://doi.org/10.5281/zenodo.10203978

[Yoo 2003] Andy B. Yoo, Morris A. Jette, Mark Grondona (2003):
SLURM: Simple Linux Utility for Resource Management.
Job Scheduling Strategies for Parallel Processing (JSSPP 2003)
Lecture Notes in Computer Science **2862**
https://doi.org/10.1007/10968987_3

[Yuen 2021] Denis Yuen, Louise Cabansay, Andrew Duncan, Gary Luu, Gregory Hogue, Charles Overbeck, Natalie Perez, Walt Shands, David Steinberg, Chaz Reid, Nneka Olunwa, Richard Hansen, Elizabeth Sheets, Ash O'Farrell, Kim Cullion, Brian D O'Connor, Benedict Paten, Lincoln Stein (2021):
The Dockstore: enhancing a community platform for sharing reproducible and accessible computational protocols.
Nucleic Acids Research **49**(W1)
https://doi.org/10.1093/nar/gkab346

[Zarras 2004] Apostolos Zarras (2004):
A Comparison Framework for Middleware Infrastructures.
The Journal of Object Technology **3**(5)
https://doi.org/10.5381/jot.2004.3.5.a2

[Zerouali 2023] Ahmed Zerouali, Ruben Opdebeeck, Coen De Roover (2023):
Helm Charts for Kubernetes Applications: Evolution, Outdatedness and Security Risks.
IEEE/ACM 20th International Conference on Mining Software Repositories (MSR), Melbourne, Australia
https://doi.org/10.1109/MSR59073.2023.00078

[Zhao 2012] Jun Zhao, Jose Manuel Gomez-Perezy, Khalid Belhajjame, Graham Klyne, Esteban Garcia-Cuestay, Aleix Garridoy, Kristina Hettne, Marco Roos, David De Roure, Carole Goble (2012):
Why workflows break – understanding and combating decay in taverna workflows.
IEEE 8th International Conference on e-Science (eScience 2012)
https://research.manchester.ac.uk/en/publications/cba81ca4-e92c-408e-8442-383d1f15fcdf
https://doi.org/10.1109/eScience.2012.6404482

[Zoubek 2021] Filip Zoubek, Martin Winkler (2021):
RO Crates and Excel.
https://github.com/e11938258/RO-Crates-and-Excel
https://doi.org/10.5281/zenodo.5068950

[Žumer 2009] Maja Žumer (ed.) (2009):
National Bibliographies in the Digital Age: Guidance and New Directions.
IFLA Series on Bibliographic Control, IFLA Working Group on Guidelines for National Bibliographies
https://doi.org/10.1515/9783598441844

[Åkerström 2024] Wolmar Nyberg Åkerström, Kurt Baumann, Oscar Corcho, Romain David, Yann Le Franc, Bénédicte Madon, Barbara Magagna, András Micsik, Marco Molinaro, Milan Ojsteršek, Silvio Peroni, Andrea Scharnhorst, Lars Vogt, Heinrich Widmann (2024): **Developing and implementing the semantic interoperability recommendations of the EOSC Interoperability Framework**.
Zenodo, EOSC-A Semantic Interoperability Task Force
https://doi.org/10.5281/zenodo.10843882

Summary

FAIR Research Objects and Computational Workflows – A Linked Data Approach

This PhD thesis explores the topics of RO-Crate, FAIR Digital Objects (FDOs), and computational workflows, in order to examine how these can be implemented and integrated using Linked Data approaches—forming "FAIR Research Objects".

The background covers the evolution of the Semantic Web, Linked Data, and FAIR Digital Objects, which are then evaluated against the FAIR principles (Findable, Accessible, Interoperable, Reusable) and several frameworks, to consider these technologies as potential middleware for a global distributed object system. The positive outcome shows that it is possible to achieve the ultimate goal of machine-actionable research outputs.

This work introduces the broader community-developed method *RO-Crate* for packaging research artefacts with their contextual information, relationships and metadata—using Linked Data standards that have been simplified and documented in detail for easier adaptation by software developers. The tension between freedom for implementations and rigidity of semantic constraints is explored, and demonstrated by various profiles of RO-Crate that have been implemented across research domains such as bioinformatics, regulatory sciences, biodiversity and digital humanities.

Computational workflows, commonly used by scientists for reproducible data analysis across execution platforms, are then examined as potential FAIR Digital Objects. Workflows are considered as shareable research outputs (by capturing the computational method for later reuse) and as part of provenance of computational results, captured in a profile of RO-Crate. Additionally the concept of *Canonical Workflow Building Blocks* is introduced as a method for FAIR sharing of tools across different workflow systems. A case study from natural history museums and biodiversity shows how the combination of workflows and RO-Crate can be used to annotate digitised specimens step by step, and gradually build reproducible domain-specific FDOs.

The discussion part of this thesis explores how the emerging ecosystem of FAIR Digital Objects can build on the results from the collaborative development of RO-Crate to carefully adapt "just enough" of Linked Data technologies with a balance of flexibility and predictability. Future directions for RO-Crate are examined, including new adaptations and further alignments with FAIR and FDO principles. Lessons from computational workflows further inform directions of FDO and RO-Crate.

The main findings of this thesis conclude that Web approaches can achieve the goals of FDO, by using existing standards with sufficient constraints that gives developers predictability and necessary flexibility. The lightweight Linked Data recommendations of RO-Crate are shown to be implementable for a range of applications, supporting advancement of the FAIR principles through practical and interoperable use of Web standards.

Sammenvatting

FAIR Onderzoeksobject en computationele workflows – een Linked Data-aanpak

Dit proefschrift verkent hoe de onderwerpen RO-Crate, FAIR Digital Objects (FDO) en computationele workflows gecombineerd kunnen worden en te onderzoeken hoe deze kunnen worden geïmplementeerd en geïntegreerd met behulp van Linked Data-benaderingen, zodat we uitkomen bij ”FAIR Onderzoeksobject”.

De achtergrond behandelt eerst de evolutie van het Semantisch Web, Linked Data en FAIR Digital Objects, welke vervolgens worden geëvalueerd aan de hand van de FAIR-principes (F: vindbaar, A: toegankelijk, I: interoperabel, R: herbruikbaar) en verschillende andere raamwerken als potentiële middleware voor een wereldwijd gedistribueerd objectensysteem. Een positieve evaluatie geeft aan dat het mogelijk is onderzoeksresultaten machinaal te hergebruiken, het uiteindelijke doel.

Dit onderzoek introduceert de door een bredere gemeenschap ontwikkelde methode *RO-Crate* voor het verpakken van onderzoeksartefacten met hun contextuele informatie, relaties en metadata, waarbij gebruik wordt gemaakt van Linked Data-standaarden die zijn vereenvoudigd en in detail gedocumenteerd voor gebruik door softwareontwikkelaars. De spanning tussen flexibiliteit voor implementaties en de rigiditeit van semantische beperkingen wordt onderzocht en gedemonstreerd door verschillende profielen van RO-Crate die zijn geïmplementeerd in onderzoeksgebieden zoals bioinformatica, regelgevende wetenschappen, biodiversiteit en digitale geesteswetenschappen.

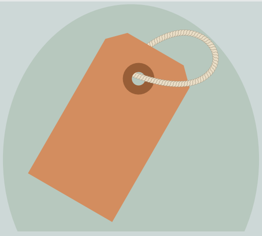
Computational workflows, die vaak worden gebruikt door wetenschappers voor reproduceerbare gegevensanalyse over uitvoeringsplatforms, worden vervolgens onderzocht als potentiële FAIR Digital Objects (FDO). Hierbij worden ze beschouwd als deelbare onderzoeksresultaten (waarbij de rekenkundige methode voor later hergebruik wordt vastgelegd) en als een weergave van de herkomst van berekende resultaten vastgelegd in een RO-Crate-profiel. Daarnaast wordt het concept van *Primaire workflow-bouwstenen* geïntroduceerd als een methode voor FAIR-delen van rekeninstrumenten over verschillende workflowsystemen. Een casestudy uit natuurhistorische musea en biodiversiteit laat zien hoe de combinatie van workflows en RO-Crate kan worden gebruikt om gedigitaliseerde specimens stap voor step te annoteren en reproduceerbare, domeinspecifieke FDO's op te bouwen.

De Discussie bespreekt hoe het opkomende ecosysteem van FAIR Digital Objects verder kan bouwen op de resultaten uit de gemeenschappelijke ontwikkeling van RO-Crate en zorgvuldig ”net genoeg” van Linked Data-technologieën kan hergebruiken, waarbij flexibiliteit en voorspelbaarheid in evenwicht worden gebracht. Toekomstige richtingen voor RO-Crate worden besproken, waaronder nieuwe adaptaties en verdere afstemming met FAIR- en FDO-principes. Lessen uit computationele workflows informeren ons verder over de richtingen van FDO en RO-Crate kunnen nemen.

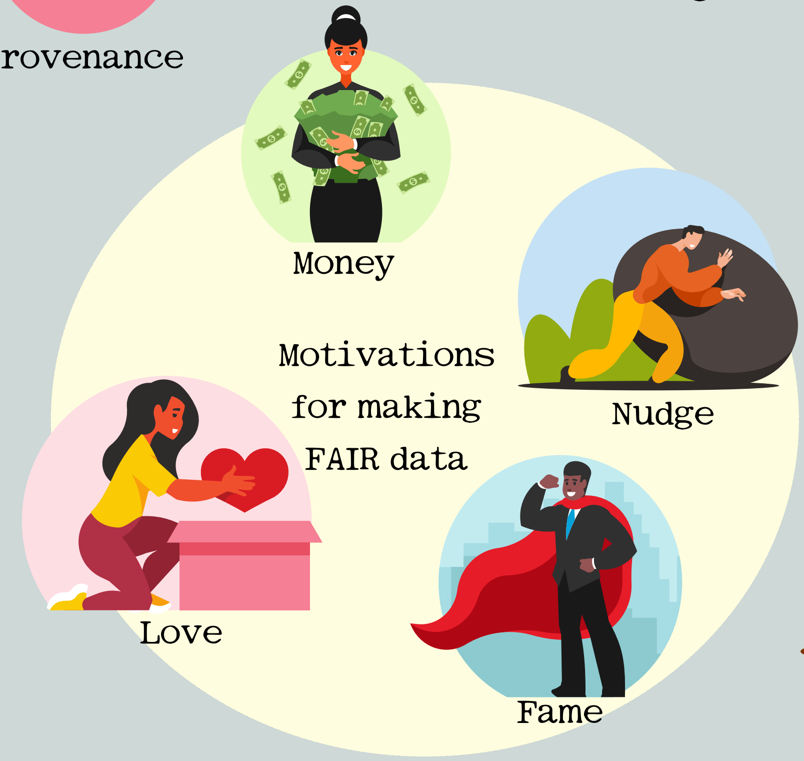
De belangrijkste bevindingen van dit proefschrift concluderen dat webstandaarden de doelstellingen van FDO kunnen bereiken, door gebruik te maken van bestaande standaarden met voldoende beperkingen die ontwikkelaars voorspelbaarheid en de nodige flexibiliteit geven. De lichtgewicht Linked Data-aanbevelingen van RO-Crate blijken implementeerbaar te zijn voor een reeks toepassingen, waarbij de vooruitgang van de FAIR-principes wordt ondersteund door praktisch en interoperabel gebruik van webstandaarden.

Provenance

Content sniffing


Metadata

Digital Specimen



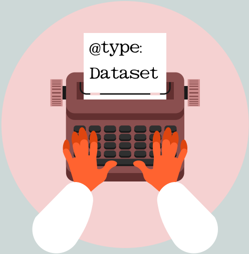
BagIt

Data lake

Tombstone

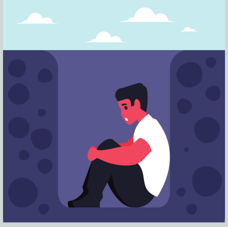
Research Software Engineer

Community



Types

ORCID


Valley of despair

"You can download our code from the URL supplied. Good luck downloading the only postdoc who can get it to run, though"

UNIVERSITY OF AMSTERDAM
Informatics Institute

MANCHESTER
1824
The University of Manchester

elixir

Organisations

cc BY

License

ISBN: 978-9-49339-178-9
9 789493 391789

Persistent identifier