Schwardmann, U. 2020. Digital Objects – FAIR Digital Objects: Which Services Are Required? *Data Science Journal*, 19: 15, pp. 1–6. DOI: https://doi.org/10.5334/dsj-2020-015

PRACTICE PAPER

Digital Objects – FAIR Digital Objects: Which Services Are Required?

Ulrich Schwardmann
Gesellschaft für wissenschaftliche Datenverarbeitung Göttingen (GWDG), Göttingen, DE
uschwar1@gwdg.de

Some of the early Research Data Alliance working groups reused the notion of digital objects as digital entities described by metadata and referenced by a persistent identifier. In recent times the FAIR principles became a prominent role as framework for the sustainability of scientific data. Both approaches had always machine actionability, the capability of computational systems to use services on data without human intervention, in their focus. The more technical approach of digital objects turned out to provide a complementary view on several aspects of the policy framework of FAIR from a technical perspective. After a deeper analysis and integration of these concepts by a group of European data experts the discussion intensified on so called FAIR digital objects. But they need to be accompanied by services as building blocks for automated processes. We will describe the components of this framework and its potentials here, and also which services inside this framework are required.

Keywords: digital objects; FAIR; data type; data type registry

Necessary Abstractions in the Data Domain

Several studies in relevant data analytic projects, for instance a survey of RDA Europe (RDA Europe 2019) from 2013, say that up to 80% of the time of experts working with data is wasted with data wrangling (i.e. making data ready for analytics). This suggests that only a high degree of automation based on simple structures can provide an alternative to this highly inefficient and error prone way of data handling.

The major obstacle for automation is the heterogeneity and complexity of data and abstraction is a generic way to hide this heterogeneity and complexity by encapsulation and virtualization.

By **encapsulation** details are hidden that are not needed at a specific layer. For instance at the data infrastructure layer there is no difference to be made between data, metadata, software, semantic assertions etc. All can be seen as some kind of data, for example as files in a filesystem, that is copied, changed or deleted. At that layer all operations do not distinguish between metadata and data, whereas on a data management and reuse layer a distinction is necessary and metadata must be used to govern the management operations on data.

By **virtualization** one substitutes objects by their logical representation. The most abstract way of such a logical representation is the pointer that leads to the object, a classical and often used approach in Computer Science, hiding all complexity behind a pure reference to the object. With Virtual Machines for instance as another virtualization example one hides only the hardware, but still exposes most of the internal structure in the logical representation.

Digital Objects

A first step of abstraction, thus virtualization and encapsulation, of data is the identification of minimal elements that are to some degree atomic from the perspective of data management and reuse. Already more than twentyfive years ago these elements have been called digital objects, as described in a reprint of an article from 1995 (Kahn, Wilensky 2006), which was reused and adapted by the RDA (RDA 2019) working group on "Data Foundations and Terminology" (Berg-Cross 2015). They can be thought as some generalization

of files in local file systems or streams of streaming providers for instance, and they are embedded in a structure of other important data concepts as one can see in **Figure 1** below.

How to represent the logical structure of digital objects with the right level of abstraction however, is in its details still a matter of discussion. As we have seen before, it certainly depends on how much of the logical structure is hidden by encapsulation behind a certain layer. And it also will partly depend on the data itself, specific workflows and use cases of data management and reuse.

But in any case the pointer as the most abstract logical representation has a prominent role here, and since data is and must be available across domains and sites, the pointer has to be a reference that is globally unique.

A global reference as URL could be seen as the easiest option, but URLs are unpredictable unstable references, because they change if the location of the data changes. See also (Klein et el. 2014) for a deeper analysis of this problem and its consequences for scientific reproducibility. This problem is known by librarians since many years and it is somehow documented in the name shelf mark that came originally from the mark for the location of a book in the shelf. After a short time it turned out that it does not make sense to always place the book at the same location. A level of redirection was introduced and shelf marks became symbolic entries in a catalog.

This additional level of redirection essentially is the rationale behind persistent identifiers. They are just globally unique strings without any semantics, but each such string has a record in a database that leads to the object, for instance via URL. If it changes, the database record can and has to be changed. These PIDs are seen as the right way to reference objects. The service to get the path to the object from the identifier string as reference, is called resolution. And since we are dealing with global references, they need to be globally resolvable, meaning that there must be a simple globally organized way that leads to the referred object. Otherwise these references would not be pointers in the sense of a logical representation in Computer Science.

Luckily such persistent identifiers are already widely used as global references in several domains of data management and publication and different highly reliable, global infrastructures are available since many years. Most of these PID infrastructures do not provide by themselves global resolution, but one of these proven systems, the Handle system (Handle 2019), has an inherent, highly scalable global resolution mechanism. Therefore the PIDs of the Handle system are able to actually fulfill the role of pointers as logical representation of digital objects.

The FAIR Digital Object Framework

The FAIR principles

The FAIR approach has been defined much later, about three years ago, as "Data and services that are findable, accessible, interoperable, and re-usable both for machines and for people" articulated by fifteen high-level principles (Wilkinson et al. 2016). These FAIR principles have already become part of the EOSC roadmap as one can see in (European Commission 2018). Most of these principles repeat on a high level view again the strong relationship between metadata, the data or digital object itself and the persistent identifier as already described in **Figure 1**. But they go even further by stating for instance that metadata has to specify

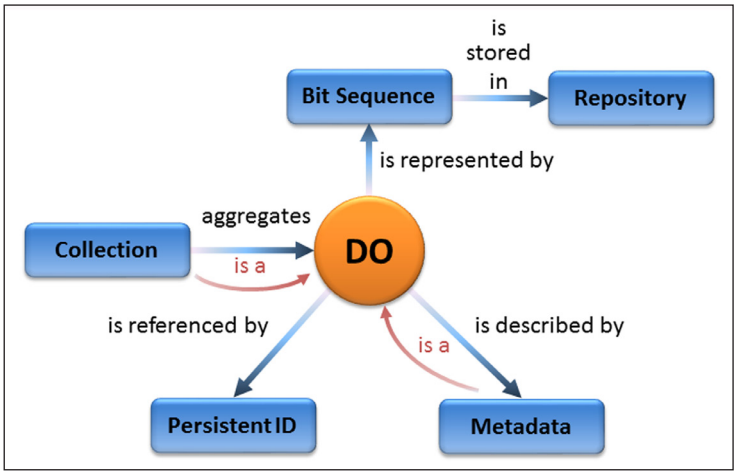

Figure 1: The Digital Object (DO) embedded in a structure of other important data elements and concepts.

the data identifier (see F4 of the FAIR principles) and (meta)data are retrievable by their identifier using a standardized communications protocol (see A1 of the FAIR principles).

This shows on one hand the strong coupling between digital objects and the FAIR principles, but the approaches are conceptually on completely different levels on the other hand: the FAIR principles are policies, whereas the digital objects are technical abstractions. This together suggests that a deep interconnection of both approaches can be extremely fruitful, because concrete implementations of digital objectss will lead to data structures that implicitly comply to at least parts of the policies. The idea, to investigate this coupling more deeply and to describe FAIR Digital Objects (FDO) as digital objects that fulfill all FAIR principles, was beside others conducted by the GEDE Digital Object Topic Group of European Data Experts (GEDE 2019) and is also described in (Schultes 2018).

Persistent Identifier, Handles and DOIs

As mentioned before the persistent identifier as pointer plays a prominent role in the abstraction process as well as in the FAIR principles and therefore in the FDO framework. Additionally there are clear advantages to use the Handle system as PID technology to describe FDOs. The so called digital object identifiers (DOI) by the way, mainly used for the publication of articles or data, are also Handles with certain additional policies. But Handles can have a much broader scope and the policies, which are necessary for publications, are not always flexible enough to fulfill the needs of data management or data sharing between researchers. For data management or data sharing usually digital content related or community specific information, often in a finer granularity and often in a tight connection to the reference, is much more important than bibliographic information. Therefore there is a need for other governance structures for Handles to ensure reliable PID services with a much higher flexibility in PID usage and policies.

Data Types

In addition to the virtualization by reference it is crucial to provide a description of the object that is understandable also by machines in order to overcome the highly inefficient current way of data handling and to choose and prepare flexible services for digital objects in scientific workflows. And it would be helpful, if these descriptions would be available already at the reference level.

One already knows this principle from the simple characterization of digital objects via MIME types, where the ending of a reference URL gives the necessary information. But for the reusability of data a lot of other and more refined parameters are necessary. Such metadata enhancements of the digital objects are called *data types.*

As mentioned before the FAIR principles as well as the notion of the digital object emphasize a close coupling between metadata, data and the persistent identifier as pointer. With the abstract structures of the FDO this becomes more explicit. Already in the early RDA working groups “PID Information Types” and “Data Type Registries” the coupling was made even tighter by allowing certain kinds of metadata to become part of the identifier record in the resolution database. Such metadata is called *PID information type* and build, as shown in **Figure 2**, a substantial encapsulation of complexity into a generic structure.

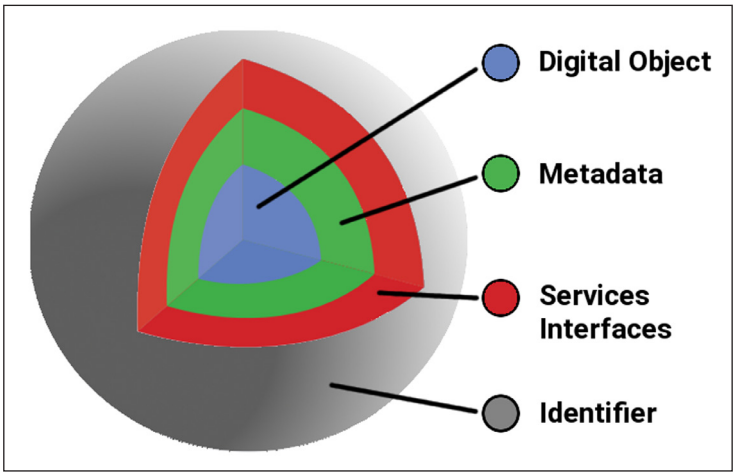

Figure 2: Encapsulation of the digital object, metadata and interfaces for services into a single logical element referenced by a persistent identifier.

But one has to choose these additional metadata elements in the PID record carefully, because an extensive use of additional fields might slow down the resolution infrastructure. So an additional RDA working group on "Kernel Information Types" developed rules and a profile (Weigel 2018) for a set of simple and most frequently needed metadata elements that should be stored together with the PID. The profile can be extended for the needs of scientific communities for instance and the rules are guidelines for these extensions. Currently this powerful technology is not supported by the DOI providers for paper and data publication. For scientific data management it is available with the more general Handle system, as provided for instance by ePIC.

Data Type Registries

In any case these types need some kind of standardization to fulfill a minimal level of interoperability, another major goal of the FAIR principles. The classical way along the procedures of international standardization bodies is either too specific or not flexible and fast enough to cover the needs of diverse research and economic areas in this fast growing area of data management.

A more promising approach is to provide community driven, reliable registries that contain reviewed type definitions in machine readable and interpretable form, uniquely referred and disambiguated again by PIDs. The PIDs of the type definitions can be used as keys for the metadata relevant to the Digital Objects as value, either in the PID record or a special metadata record.

Such registries with type definitions are called Data Type Registries (DTRs) and have been a topic for the Research Data Alliance (RDA) also since its first days (Lannom 2015). Two working groups made recommendations that led to a prototypical deployment of a working DTR implementation based on Cordra. Cordra is an open source software for managing digital objects, now available in version 2.0. ePIC is running two instances of Cordra, configured as DTRs on behalf of ePIC, one for production data types and one for the preparation of data types and testing. The type definitions are openly available. To create or change types an account is needed. A distinctive feature of the ePIC DTRs is the ability to define types in a hierarchical manner, such that also complex data types can be easily defined and for instance schemata for the value domain can be derived from the definition (Schwardmann 2016). As a starting point one can find a short overview with links to these DTRs at the ePIC web pages (ePIC 2019).

Because DTRs enable the disambiguation and correct assignment of types for humans and machines, they build an integral part of the FDO framework. With the correct choice of PID information types, depending on the needs in a scientific community, such FDOs enable fast decisions at the reference level about the relevance of data for certain scientific questions, allow the identification of the location and prepare the automated staging of remote data for the processing in a scientific workflow, for instance with high performance computing, or even the automated decision that a remote computation would need less effort.

Which Services are required?

The introduction of PIDs as reliable pointers or references to digital objects is a precondition for long term findability and provides already additional simplification and flexibility in the data domain. As mentioned in the beginning, the major goal is the enabling of automation, and especially for findability essential requirements for automating data findability were given in (Weigel 2020). There are several elementary services for PIDs like creating, managing and resolving them. Also basic services can be used on PID records, if they contain additional metadata as PID information types.

Examples here are the detection of duplicates based on checksums, of earlier versions based on 'was derived from' relations or of the candidates for format conversion based on mime types and version numbers. A metadata service based on the metadata location given in a PID information type would be another example. Also decisions in workflows can be based on such PID information types as for instance the decision to move the application to the data or the data to the application based on the data size.

Collection representations can be based completely on PID information types, and a wide range of additional services and applications are proposed as part of the collection API and also beyond. Furthermore for repository interoperability it would be beneficial to provide a collection enhancement based on a common agreement as it was given by the RDA working group on Research Data Collections (Weigel 2017) to enable more flexibility for structures imposed on digital objects.

All these examples show that the elementary service of resolution for retrieving PID information types from the PID record is required, but also services to describe the types in data type registries and to retrieve this information are needed. This additionally asks for interoperability between DTRs and services that

monitor this interoperability. And in a next step services are required, that provide a set of information types that can be expected from (a class of) PIDs, so called PID profiles.

Repositories

But finally the data services for FDOs itself need to be based on repositories providing reliable access to elementary digital objects. Currently often these repositories are giving some data representation enhanced with data base systems that provide a local layer of data and metadata indexing. A PID registration for the provided data is not even given too often. A FAIR and global data perspective proposes a clear statement to overcome this situation. The FDO has to replace all other kind of representation of data inside repositories and a more generic approach to metadata indexing is also necessary. In some cases it will be possible to provide adapters around legacy repository architectures, but overall this transformation is a big effort and may take a while. Nevertheless this effort is worthwhile in order to not end up with a fragmented data space with all its interoperability gaps, as we have it today.

Competing Interests

The author has no competing interests to declare.

References

Berg-Cross, G, Ritz, R and **Wittenburg, P.** 2015. Core Term Definitions. In: *Data Foundation and Terminology Work Group Products*. DOI: https://doi.org/10.15497/06825049-8CA4-40BD-BCAF-DE9F0EA2FADF

ePIC. 2019. *ePIC Persistent Identifiers for eResearch*. Available at http://dtr.pidconsortium.net/ [Last accessed 9 December 2019].

European Commission: Directorate-General for Research and Innovation. 2018. Turning FAIR into reality: Final report and action plan from the European Commission expert group on FAIR data. DOI: https://doi.org/10.2777/1524

GEDE. 2019. *Group of European Data Experts*. Available at https://rd-alliance.org/group/gede-group-european-data-experts-rda/wiki/gede-digital-object-topic-group [Last accessed 9 December 2019].

Handle. 2019. *The Handle System*. Available at http://www.handle.net [Last accessed 9 December 2019].

Kahn, R and **Wilensky, R.** 2006. A framework for distributed digital object services. *Int. J. on Digital Libraries*, 6: 115–123. DOI: https://doi.org/10.1007/s00799-005-0128-x

Klein, M, Van de Sompel, H, Sanderson, R, Shankar, H, Balakireva, L, Zhou, K and **Tobin, R.** 2014. Scholarly Context Not Found: One in Five Articles Suffers from Reference Rot. *PLoS ONE*, 9(12): e115253. DOI: https://doi.org/10.1371/journal.pone.0115253

Lannom, L, Broeder, D and **Manepalli, G.** 2015. RDA Data Type Registries Working Group Output. DOI: https://doi.org/10.15497/A5BCD108-ECC4-41BE-91A7-20112FF77458

RDA. 2019. *Research Data Alliance*. Available at https://rd-alliance.org [Last accessed 9 December 2019].

RDA Europe. 2019. *Research Data Alliance – Europe*. Available at https://www.rd-alliance.org/rda-europe [Last accessed 9 December 2019].

Schultes, E and **Wittenburg, P.** 2018. FAIR Principles and Digital Objects: Accelerating Convergence on a Data Infrastructure. In: Manolopoulos, Y and Stupnikov, S (eds.), *Data Analytics and Management in Data Intensive Domains. DAMDID/RCDL 2018. Communications in Computer and Information Science*, 1003. Cham: Springer. DOI: https://doi.org/10.1007/978-3-030-23584-0

Schwardmann, U. 2016. Automated schema extraction for PID information types. *2016 IEEE International Conference on Big Data*. PID:21.11101/0000-0002-A987-7. DOI: https://doi.org/10.1109/BigData.2016.7840957

Weigel, T, Almas, B, Baumgardt, F, Zastrow, T, Schwardmann, U, Hellström, M, Quinteros, J and **Fleischer, D.** 2017. Recommendation on Research Data Collections, RDA. DOI: https://doi.org/10.15497/RDA00022

Weigel, T, Plale, B, Parsons, M, Zhou, G, Luo, Y, Schwardmann, U, Quick, R, Hellström, M and **Kurakawa, K.** 2018. RDA Recommendation on PID Kernel Information (Version 1), RDA. DOI: https://doi.org/10.15497/RDA00031

Weigel, T, Schwardmann, U, Klump, J, Bendoukha, S and **Quick, R.** 2020. Making data and workflows findable for machines. *Data Intelligence*, 2: 40–46. DOI: https://doi.org/10.1162/dint_a_00026

Wilkinson, MD, et al. 2016. The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data*, 3. DOI: https://doi.org/10.1038/sdata.2016.18

How to cite this article: Schwardmann, U. 2020. Digital Objects – FAIR Digital Objects: Which Services Are Required? *Data Science Journal,* 19: 15, pp. 1–6. DOI: https://doi.org/10.5334/dsj-2020-015

Submitted: 13 December 2019 **Accepted:** 09 March 2020 **Published:** 01 April 2020

]u[*Data Science Journal* is a peer-reviewed open access journal published by Ubiquity Press.

OPEN ACCESS