Digital Objects as Drivers towards Convergence in Data Infrastructures

Peter Wittenburg (Max Planck Computing & Data Facility, Garching/Munich)
George Strawn (US National Academy of Sciences, Washington)
Barend Mons (GO FAIR International Support and Coordination Office, Leiden)
Luiz Bonino (GO FAIR International Support and Coordination Office, Leiden)
Erik Schultes (GO FAIR International Support and Coordination Office, Leiden)

December 2018

Corresponding author: Peter Wittenburg (peter.wittenburg@mpcdf.mpg.de)

http://doi.org/10.23728/b2share.b605d85809ca45679b110719b6c6cb11

Executive Summary

In a recent paper Wittenburg & Strawn argue that it is time to come to convergence in the domain of data, to overcome the huge inefficiencies of data wrangling and to create a phase of rich exploitation of data. Such a converging data domain needs to be based on stable and simple specifications and would drive future scientific endeavor and economic growth. In this paper the authors look in more detail at the concept of Digital Objects (DO) from a technical perspective, and we analyze their possible contribution to achieving this urgently required convergence. DOs capture meaningful content of different kinds (data, metadata, software, digital representations of physical objects, etc.), which scientists want to exchange, combine and analyze. While industry currently tackles these challenges mainly by developing complex proprietary frameworks or by comprehensive reference architectures, the authors propose a component-based and bottom-up approach. Moreover, DOs are a practical and scalable way to facilitate the implementation of the FAIR principles, which should guide the progress of this bottom-up approach.

We argue that DOs should be the primary components of a future stable domain of digital entities, and they need to have clear identities including unique and globally resolvable persistent identifiers (here referred to as PID's), metadata describing their various properties, and a set of associated operations. This paper makes a number of assertions about the value of DOs:

- The definition of DOs allows us to implement abstraction and encapsulation methods, which have been proven to be extremely powerful when they have been used in the designing of other complex systems and could benefit the eco-system of data infrastructures that will likely be highly complex as well.
- The persistent identifiers crucial for implementing a DO domain allow creating stable pointers to all relevant components of each DO, such as the different kinds of metadata needed to support findability, accessibility, interoperability and reuse. The Handle System, governed by the Swiss DONA Foundation, is such a global PID resolution system.
- Due to the capabilities of DOs they are not only a useful technical concept, but allow scientists to design a stable domain of digital entities that has the desired properties to be sustained for the next century, which is appropriate given the huge investments that will be made toward data.
- DOs are an excellent basis to develop a stable eco-system of variable and dispersed data infrastructures, since they will solve interoperability challenges, which cause most of the inefficiencies today at the data organization layer.

- The DO Interface Protocol (DOIP; https://www.dona.net/doipv1doc), which is based on ITU X.1255, will have a normalizing effect across the heterogeneous repositories that can be compared to TCP/IP internetworking across different island networks decades ago.
- DOs provide abstraction, binding and encapsulation mechanisms, the latter allowing the association of operations with classes of DOs, and thus make a perfect starting point to orchestrate automatic workflows and to tackle the reproducibility challenge.
- DOs offer enhanced data protection possibilities since protected PID attributes of a clearly identified DO could be used as a secure point to access permission records and to blockchain entries that contain smart contracts (i.e. machine actionable licenses) and transaction records.

The complementarity of the FAIR guiding principles and the DO concept as a way to implement FAIR and the increasingly close collaboration between initiatives such as the Research Data Alliance, the GO FAIR initiative, CODATA, the DONA Foundation and domain science organizations provide an excellent basis to further develop the DO concept with the help of scientific use cases and testbeds. This could also be a crucial contribution to the realization of international ambitions such as the European Open Science Cloud.

1. Introduction

In their recent paper on common patterns in the development of global infrastructures, Wittenburg & Strawn [1] summarized a few facts indicating that the data domain in science and industry suffers from huge fragmentation at all levels. A proliferation of tools that meet the detailed needs of domain users, have as a consequence an extreme inefficiency in data driven projects where about 80% of the work is simply wasted with data wrangling. Many current projects fail due to the required efforts and many experts are excluded from data intensive work. They compare the situation in the data domain with other domains that went through similar phases of what they call “creolization". They point to a few global initiatives to find common drivers of convergence and thus conclude that the time is ripe to come to revolutionizing agreements as happened in the other investigated domains. Their conclusion is that the concept of Digital Objects can change our practices fundamentally.

From recent discussions at expert workshops [2, 3] we know that big IT industry is observing similar phenomena but are under pressure to satisfy their customers' short-term needs, i.e. coping *ad hoc* with fragmentation and proliferation is part of their business model. Frameworks and platforms are being designed based on proprietary standards that have adapters to all kinds of formats and tools to enable data, and tool, integration. The authors take a different view since we are convinced that we now need to lay the basis for proper, scalable and sustainable data management and stewardship of digital resources for the next centuries. Digital data will be as relevant a part of our cultural memory as books were for our memory for thousands of years. We urgently need to define drivers towards convergence as a fundamental step towards a scalable and sustainable data infrastructure, a step which will require a paradigm shift from *ad hoc* solutions to approaches that have the potential to stay operational for at least the following decades without hampering the freedom of technological innovation. There is no way out of fundamentally rethinking data management and access if we want to come to a stable data domain that indeed is forming a powerful analytics ecosystem and a digital memory of our times. This paradigm shift is the reason why we do not see the Web, which has been a very successful way of exchanging information and processing it, as the solution for managing the rapidly increasing amounts of data for the future, nor as a way to avoid the digital Dark Age as described by some experts [4].

The FAIR principles [5] are an excellent first step towards convergence, but they are not a detailed guide to infrastructure building and technology development. Initiatives such as the Research Data Alliance [6], C2CAMP [7] and GO FAIR [8] are internationally active drivers of progress due to their bottom up nature of organization. While RDA is focusing on specifications of guidelines, procedures, components and their interfaces, GO FAIR is focusing on implementing the FAIR principles in three directions: awareness and policy, education, and technology. Many of the outcomes of RDA will be road tested in GO FAIR implementation networks. For instance, C2CAMP, consisting of experts who were involved in the birth of RDA, form an example of an effective bridge between the design and implementation, since they want to implement a testbed of several RDA results focusing on Digital Objects and act as an implementation network within GO FAIR.

In this paper on the role of DO's, we want to take a largely technical point of view, i.e. (1) look in more detail at related developments in computer science and information technology and identify their major contributions in the above sense, (2) describe briefly that DOs are not just an implementation concept but also a way to conceptionally organize the digital domains in science[1], and (3) describe in more detail the basics of the concept of Digital Objects.

2. Early Evolution of the Concept of Digital Object

According to Wikipedia and other resources [9, 10] the Latin word "objectus" can be seen as the source for our current use of the word "object". It is said to be "something to throw or to put before or against". In modern philosophy (Descartes, etc.), the contrast between "subject" and "object" was introduced where the "subject" is an observer and the "object" the thing to be observed, i.e. the meaning of the original Latin word was extended considerably. Now the "object" could also include abstract entities instead of physical entities only. Basically, this is the meaning of the word "object" we are currently using and which can be applied to our world of distinguishable entities - be they physically existing or not, concrete or abstract, or be they digital representations of such entities. Philosophy [11] tells us that such entities have properties which can be attributes or exist as components of them.

Since we want to communicate about such "objects", do something with them, and refer to them, they must be meaningful in a certain context, have names and properties. In the case of mass-objects we tend to give names to the **class** of the objects, such as dollar bill, but each individual bill still has a number to be able to uniquely identify it, since in certain contexts each individual object can become meaningful. Such names need to be defined in the name space reserved by a certain authority. In general, **properties** describe a class of objects and not an individual object, although in specific contexts the properties of individual objects could be of relevance.

"Digital objects" are therefore meaningful entities that exist in the world of bits which includes all kinds of entities humans or machines need to uniquely identify, such as data, metadata, software code, queries, configurations, etc. being available in digital form. This also includes digital representations of physical entities such as persons, institutions and abstract entities such as concepts and relations. Therefore, Digital Objects (DOs) need to have some **content** (represented by a bit sequence stored in some repositories), an **identifier,** and **properties** of different **types**. Since Digital Objects are so numerous and increasingly need to be dealt with by computers, the lingual 'names' are not sufficient anymore, although in some communities "reference collections", for example, are referenced by their name in human communication. For DOs, we can state that they are central not only for unambiguous human communication, but increasingly often also for machine to machine communication and therefore we need to be able to identify them properly and most important at all: unambiguously.

We can summarize that "objects" are meaningful entities that are central in human communication and interaction and that they have names (identities) and properties to uniquely

[1] A more detailed analysis of the role of DOs for data science is required.

refer to them. Digital Objects are meaningful entities in the digital domain having names (identities) and properties as well.

The term "Digital Object" was used by Kahn and Wilensky in their 1995 paper [12] as a response to something essentially missing in the Internet. The Internet specifies devices that exchange basically meaningless messages (i.e., sender and receiver must understand the meaning by methods external to the message itself). At the sender side entities are chopped into fragments and routed through a network of nodes to the receiver which then puts the fragments together using standard protocols such as TCP. Eventually, users/clients want to exchange meaningful entities of different types. This is why in the early Internet protocols such as FTP (1971) were specified to enable the exchange of file information via the Internet. In 1989 the very successful HTTP protocol was added to exchange basically HTML encoded information, offering new possibilities compared to FTP. HTTP is an application layer protocol for distributed, collaborative and hypermedia information systems relying on the HTML standard and thus founding the World Wide Web. All sorts of techniques and methods were invented to make the Web a generic platform to exchange all kinds of information where URIs in the form of URLs were used to address the information. However properly defined and ‘non-decomposable’ DO’s still were not an integral part of this global infrastructure.

In a revised version of that early paper in 2006 [13], Kahn and Wilensky describe a domain of Digital Objects as a generalization of what is indicated in figure 1: Digital Objects are the meaningful entities exchanged between two actors (i.e., entities that describe their meaning independent of any understandings between the two actors), and there is a protocol that allows any client to interact with such DOs. In their earlier papers their approach was called Repository Access Protocol. In 1997 CNRI organized the Cross-industry Working Team with various experts from major companies and a paper in 1997 supporting the concept of DOs was produced [14]. However, subsequently, industry first concentrated on exploiting the emerging Web paradigm as a means to respond to the increasing demands from its customers who made use of the new Web opportunities.

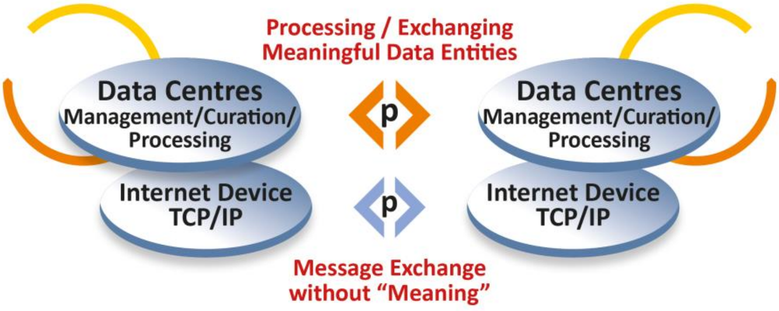

Figure 1 indicates the two layers of exchange in the Internet. At the bottom level is the datagram exchange between Internet Devices based on TCP/IP. At the top level is the exchange of meaningful information which needs to be supported by a dedicated protocol.

As already indicated, many people use the web for "managing" and in particular exchanging the increasing amount of data and information. What we are reflecting on in this paper is whether we need a new, consolidated paradigm for data management and stewardship[2] given the increasing amount and complexity of data and the necessary shift to automatic procedures. The web gave us a way to point at all manner of remote files and databases, send commands and requests to those data sources, and build pockets of networked information management. Now we need to consolidate these gains and normalize the interfaces to improve our understanding of the large amount of scientific and technical data that is now available on the internet and the even larger amount that is coming at us in waves in the near future. Experience shows that most research domains and repositories dealing with large data sets [15] started to use PIDs in form of Handles to identify each entity of relevance and to associate immediately relevant information (metadata) with it, i.e., we can

[2] In this paper we see data management and data stewardship as synonyms despite some semantic differences between these two terms.

see that a new way to establish stability of references has been chosen by various communities and increasingly these are consolidated in approaches recommended by, for instance, RDA and CODATA and implemented in research infrastructures such as GO FAIR implementation networks.

Major arguments as to why we cannot see the Web as a ready-to-use solution in the above sense can be summarized as follows: (1) Identification of digital entities needs to be stable and independent of protocols, that will change over the decades, i.e. identifiers need to be independent of ephemeral aspects of the entity being referenced such as location, ownership as indicated by the domain name, and usage of a specific database schema; (2) the Web, by design, is ephemeral and we are used to this characteristic of it, i.e. its content is continuously changing and web-archives can only be seen as an interim solution to overcome its main disadvantages; (3) the Web certainly suffers from a lack of explicit and consistently applied security and quality assessment features, i.e., measures are not intrinsically associated with the data itself, but dependent on the data providers; and (4) the Web is basically a delivery system where the local organization of data and the mechanisms of the Web are completely decoupled, which can be seen as one of its initial strengths. In contrast to these emerging deficits in the current protocols and approaches, DOs are entities that are independent of the way their content is stored and organized and of their location, which makes them fundamentally different to the current approaches. We can associate all kinds of attributes with DOs, however, we need to ensure that we have powerful mechanisms to find the relevant information about them, and thus binding is crucial. Some need to be closely related to their identity, comparable to the passports of persons which include immediately relevant information about the person, since otherwise identity is meaningless. This all must be stored in a secure and persistent way, such as in PID records. The Web does not have inherent facilities for this.

3. Related Work in IT

In this chapter we want to refer to some activities in computer science (IT) that are related to the concept of DOs and thus can help us to understand their place in the landscape of concepts.

Object Orientation

Experts in early artificial intelligence [16] were motivated as early as the 1960s to mimic the world of physical objects by creating (digital) "objects" exchanging "messages" which led to a new paradigm in programming. The Simula programming language, for example, introduced innovative and "object-related" concepts such as class, object, inheritance and dynamic binding. A whole range of other programming languages introduced additional concepts to support, for example, data encapsulation and messaging. In Object Oriented Programming (OOP), objects thus contain more or less complex internal data structures hidden to the user, and are instances of classes (defining their type) including the operations (methods) that can be executed to manipulate the internal state and thus allowing them to meaningfully interact with each other [17].

The binding of data structures and functions that operate on them and the restriction to only use these predefined functions leads to encapsulation, which prohibits unintended or erroneous access and changes. An advantage of this object-based approach is also that users/clients do not have to deal with the internal complexity and nature of the implementation of data structures. The available methods form an interface which users/clients need to utilize instead. The notion of classes allows abstractions in so far as properties and methods can be associated with classes (types) and thus being inherited by all objects (instances) of a class.

In close relationship to OOP, Abstract Data Types were introduced in computer science in 1974 [18, 19] as a principle to practice information hiding and thus making the programming of large complex systems more robust. Data types are defined from the point of view of a user of the data including, in particular, the possible operations on specific types, i.e. the internal structure of an object is hidden from the user and designed, implemented and tested by the developer. The user can access the internal structures only by making use of the predefined operations. Abstract Data Types were introduced as a theoretical concept and as a design principle to be followed. We have learned to use their advantages in the design and implementation of complex systems and to see them as stable islands in a highly dynamic sea of implementations. It's the systematic application of the principle of abstraction that allows developers to change underlying technology and users/clients to ignore these changes of technology. It's the interface that is relevant for them and that will change at much slower rates.

Key for the success of the new object-oriented paradigm in programming was encapsulation and the provision of a set of defined methods that allow manipulating the internal state of objects. These basic principles showed a way to manage complexity.

Object Stores

In parallel to this OOP programming paradigm change, experts worked on "object stores" where each data entity includes the corresponding bit sequences, describing metadata and an identifier unique for the given namespace [20]. The intention of the object stores was to abstract the user away from lower layer details. In doing so, it offered the possibilities of new addressing mechanisms through IDs and flexible metadata, of a way towards "unlimited" storage systems and of exchanging objects that include all relevant information to reuse it. The trade off was an additional administration layer that would house IDs and other metadata and offer the functionality to work with such objects.

The notion of Digital Objects was taken up by some data-related initiatives at early stages. Inspired by the Kahn & Wilensky paper, Cornell University and CNRI started designing and developing software for repositories to manage DOs in the 1990s. Cornell's work resulted in the FEDORA package [21], now being maintained by Duraspace. It offers ways to define DOs which can also be collections, to associate metadata with them which include PIDs, to define relationships between objects and to refer to procedures that can operate on DOs. Due to its modular functionality it is widely used to establish repositories. In 2002 MIT and HPLabs developed the D-SPACE software to also manage a repository of different content [22]. They followed a more traditional design approach by making use of relational databases to store metadata and by offering a complete software stack to run and access a repository. D-SPACE is not built on the concept of DOs, however, it did introduce the systematic use of Handles to allow managers to assign persistent identifiers to all entities, and could probably be made DO-supporting with relatively modest adaptations.

In parallel, the concept of Object Stores was investigated by a research project at CMU in 1996 [23]. The intention was not only to abstract away from lower layer storage functions, but also to bind descriptive properties (metadata) and a unique PID with each object to offer many more opportunities for data management and access. Some object store experts used "buckets" as containers for objects to add even more flexibility and capacity. Key was the possibility of associating stored entities such as files with much more metadata than was possible in traditional file systems, to introduce policies with the different objects, and to add an administration layer that maps objects to physical configurations (nodes, discs, etc.). Some file systems for very large stores did work out a variant to store metadata in databases, providing pointers to the objects' bit sequences stored in files, for example. This technique is now widely used in most supercomputer centers.

Faced with increasing demands for compute power and storage capacity, new concepts of parallelization were investigated. Foster and Kesselmann published their concept of Grid computing in 1998 [24] with the intention to bundle existing compute resources in such a way that a virtual supercomputer is being created by letting distributed computers work closely together. Grid middleware gives the impression to users that they have access to one big machine where load balancing etc. would be taken care of. The metaphor promised powerful solutions; however, in practice these types of Grids had to cope with many different architectures, software systems, and also organizational/administrational differences. However, Grid computing did bear an early promise of distributed learning, a highly desired property of a future data and analytics infrastructure. In 2006 Amazon offered its Elastic Compute Cloud which basically encapsulated Grids of parallel machines by offering a service-based interface and reducing the internal complexity: it put the Grid into a Cloud. This virtualization turned out to be very successful when it was combined with the principles of Object Stores as implemented in Amazon's Simple Cloud Storage Services (S3) [25].

Fundamental in Amazon S3 is the concept of "objects" that consist of data and metadata having unique IDs. S3 also introduces the concept of buckets which are containers to organize objects, define namespaces and optimize administrational aspects such as access rights, assigning a geographic region for storage, etc. A container could for example easily be assigned to a "user" or a mobile phone number. In S3 a complete ID specifies the bucket, a unique key in the bucket and a version. With specifying prefixes and delimiters in the key, hierarchies can be built, however, they are not primary in accessing an object. S3 thus allows the storage of objects, files, and also blocks that are, for example, often used to optimize work with large databases. However, in S3 all are "objects" administered by a highly optimized database that stores all relevant information about buckets and objects including the regional copies of buckets and thus objects which are automatically created for high availability reasons. Metadata are key-value pairs assigned by the system or the user and can be used to find and access "objects".

Other cloud storage providers increasingly follow similar strategies, i.e. a unique identifier is used to identify objects, user and systems defined metadata is being assigned to objects and a database bundles all relevant information on objects.

While many activities are focused around developing and making use of the concept "object" in the area of optimally organizing the data domain, we should not ignore two directions that are not using this concept but focus on pragmatic optimizations: data warehouses/lakes and NoSQL databases. Since the ascendance of the relational databases, data warehouses [26] have become the work horses used in businesses to carry out all kinds of business analytics based on the centrally stored data that has been integrated from many sources, i.e. the basis of a data warehouse is a large database, the structure of which is defined by a logical structure (schema). All incoming data is mapped to this structure to achieve interoperability and is maximally cleansed to yield proper outputs. This also implies that there is no formal distinction between data and metadata; they are just different types of data which can be typically accessed by executing more or less complex SQL statements often embedded in procedures. Key to the localized success of data warehouses are well-designed logical structures that also include historical data, the presence of excellent and documented business logic to define the queries to be executed, and optimal usage of database systems including ultrafast indexes. However, these strengths also present a major weakness when it comes to reasoning over differently structured data warehouses.

Data warehouses are thus, to a certain extent, huge silos of data where all incoming data and queries have to be mapped onto one predefined structure and semantic space. The varying and rapidly multiplying amounts of data we deal with today and in the coming decades, however, cannot easily be converted into the many different existing structured data regimes as would be expected in many data warehouses. The new term "data lakes" was coined [27] as a possible mitigation of this emerging bottleneck. "A data lake is a storage repository that holds a vast amount of raw data in its native format, including structured, semi-structured, and unstructured data. The data structure and requirements are not defined until the data is needed" [28] Data Lakes thus describe the typical situation science and industry will be confronted with even more in the future, where all kinds of sources (smart sensors of different types, crowd sourcing, simulations, etc.) generate vast amounts of data that experts want to integrate for specific analytics. The usual phenomena such as fragmentation and lack of interoperability at all levels must be tackled ad hoc for specific use, requiring for new strategies to make data linking, and where needed physical integration, more efficient.

Another branch of activities was directed towards designing specialized database structures, called No-SQL databases [29], to address specific data models much more directly, cope with the increasing amount of specific data in a much better way and utilize parallelism more efficiently than could be done in relational databases. Databases for "wide column", "document", "key-value", and "graph" data were developed, to just mention a few major data model types. Still, for databases, the core idea is to import all relevant data into one logical instance and to have powerful operators to do operations on that instance. Although it would be in principle possible, for example, to import sensor data including the relevant metadata into such a specialized database and to carry out operations on them, we do not see such databases as primary stores for long term data management, but as secondary stores to carry out specific operations.

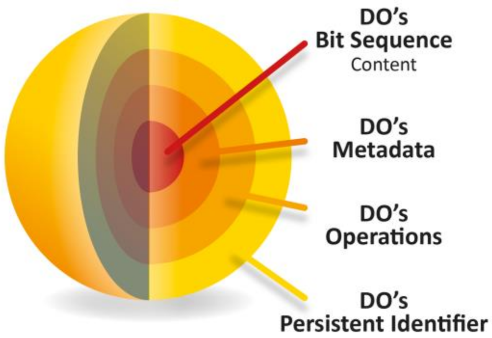

Figure 2 indicates the principles of abstraction, encapsulation and binding that are central to the concept of Digital Objects. The core of the DO is a bit sequence that is encoding some content (data, metadata, software, etc.). It is described by metadata of different kinds to enable access to and the correct interpretation of the DOs content. The DO has a PID that uniquely identifies it and its attributes point to the locations of the bit sequence, its metadata and other relevant properties, i.e. the DO's PID is opening the way to access all components of the DO including a type specification. The DO's type allows users to define type specific operations and in doing so apply encapsulation. It should be noted that metadata descriptions are themselves DOs, i.e. they are associated with a PID.

Summarizing, we can state that much work has been done to design new methods to cope with the increasing amounts and complexity of data and these can serve as a basis for repositories maintaining our data over long periods. The concept of "digital objects" as sketched graphically in figure 2 has found its way into practical use in many different areas of computer science.

Industrial Approaches

Big IT industry undertakes huge investments in bridging technology to cope with the diverging customer needs and applies all kinds of technology that fit optimally with the specific questions they are addressing. In fact, silos are being created and re-created everywhere, which turn out to be hampering interoperability when data needs to be integrated from different sources. Integration frameworks are being developed based on proprietary cores with many adaptors and much adaptation work is being done helping to enable the cross-silo data integration and to achieve interoperability between tools. IBM's “Watson” [30] and Oracle's “Universal Intermediate Representation” [31] can be seen as representatives of such approaches. Many companies deploy "platform strategies" where they define the interoperability rules and request co-operating companies to accept their specifications. However, this does not solve the fundamental problems for global data and services linkage based on opens standards sketched above.

Production industry, in establishing smart factories, takes a different route. They have been creating large consortia that worked out reference architectures (RAMI [32], IIC [33], IDS [34], etc.) with the intention to define an agreed conceptualization. They have created a terminology for the complex scenarios as described, for example, in Industry 4.0 [35], where they are able to define relevant components and their interfaces in a stepwise refinement process, and to finally create a functioning digital domain of data and processes that can preserve investments for many decades. While these initiatives start with splitting the overall complexity into modules and components in a step-down process, the basics of the concept of Digital Objects were developed from better understanding the core of data. Therefore, there is no contradiction between the two approaches, in fact, a cross fertilization is urgently required.

Summarizing, we can state that big industry is very much concerned about the inefficiencies and thus high costs in data projects. While big IT industry needs to cope with the demands of its user base and looks for fast solutions for the integration task, big production industry follows a top-down approach of stepwise refinement of specifications based on comprehensive reference architectures. Despite the high economic pressure, industry will only change and adopt new suggestions such as the DO concept if there is a chance of ensuring the return on investments.

4. Relevance of DO Structuring for Scientific Data Domains

In the two previous sections we described first the evolution of the concept of Digital Objects as a consequence of the design of the Internet and we argued that the Web, while filling a huge functional need, cannot be seen as the only solution addressing the needs of proper data management/stewardship for the future, and second that in computer science the notion of “objects” had an enormous and positive influence on mastering complexity in software design. But such elaborations do not address the question of whether this concept is relevant for science, and of whether the notion of “meaningful entity” would help science to better organize its domain of digital entities. The question to be addressed now is whether the concept of “digital objects” is simply a technical one or whether it also has a scientific dimension anticipating a development where scientists in general do not have the time to look at individual entities due to the sheer volumes, but need to use software agents to operate, inceasingly independently, on classes of entities. Are DOs a type of tool that can help us with the dilemma that most data collections will soon be beyond human comprehension? An even larger question is whether DOs are fundamental augmentations of human knowledge and reasoning, without which we will not be able to comprehend the new complexities? We will leave the last question to be answered by a follow-up paper.

Persistent Identifiers (PIDs)

Starting around 2000, many research communities began assigning persistent identifiers and metadata to their data entities [36, 37] to allow stable referencing and reuse over many decades. Recently, a collaboration of delegates from 47 different scientific communities agreed on the usage of such persistent identifiers[3] [15]. The major question addressed was the granularity of assigning PIDs to digital entities. It was concluded that the granularity depends very much on the context in which humans/machines communicate about such entities. It could be that a whole database is the "object" of reference, but it could also be that the object of reference is a part of a database invoked by a query on time-stamped data. Higher granularity can always be achieved at later stages as required, and provenance can be associated.

In many scientific communities there seem to be well established “atomic” physical entities for which it makes sense to invest effort and assign attributes. In chemistry and physics “atoms” have an outstanding role since it is possible to assign typical characteristics to the different elements and based on this predict their behavior. In linguistics we have learned that it makes sense to identify phonemes as basic elements since they can be mapped to character strings that allow us to build morpho-syntactic constructions, define lexicons, associate meanings with words as higher-level entities, etc. In biodiversity the individual specimen or molecule is a useful anchor to add relevant attributes, derive classifications, map to different name spaces, etc. All these granularity levels have been developed and stabilized over long periods of time.

[3] Most of the communities are making use of Handles in form of DOIs, ePIC Handles or others.

The digital domain is comparatively young and machine-readable metadata as well as suitable levels of granularity are not that well established. However, based on recent experiences we can define a few guidelines:

- In the digital domain it is possible to split entities into new entities and to form almost arbitrarily virtual collections from existing entities, i.e. a certain granularity choice can be revised at a later stage.
- As in the physical domain it makes sense to identify pivotal entities to associate crucial information with such digital representations of important physical entities.
- In addition, in the digital domain we have crucial entities that will be recombined to carry out all kinds of calculations such as the results of a specific experiment or observation, the results of a simulation run or the results of calculations that recombine other data to yield new results. Many different types of data could be listed here such as a video recording of a certain cultural event, an annotation of such a recording, a whole brain image from a specific person generated by an MRI scanner, a sequence of DNA generated by a sequencer, or a data sequence generated from other sensors at a specified time interval to just mention a few.
- New types of essential aggregated DOs could be derived from nanopublications and knowlets [38, 39] that emerged from an analysis in the domain of linked data

Metadata

Currently most data are still being exchanged within projects, institutions or collaborations, since this exchange can be facilitated by personal interactions about their content and format. This model works for now, but it does not scale, since for every reuse case researchers' time is required. In a situation where data reuse from other researchers gets increasingly popular other methods of exchanging "information about data" are needed. This information about data which can be of different types is called "metadata". In the digital realm the separation between metadata and the data they describe can be easily blurred, but we define as metadata any data that 'assert' something about another DO. Obviously, a set of metadata or annotations is a DO in itself and so, we can construct an ever expanding ecosystem of meaningfully interlinked DO's. DO's that are specifically created to achieve interoperability in data-intensive research have also recently been referred to as Research Objects [see ref: 66]

Researchers work in two roles: as producers and consumers. For the data producer it is important to create rich metadata so that consumers (including machines where possible) understand without personal interaction whether the data can be reused for their intention. It is important for researchers to find potentially relevant data with the help of tools and gateways that offer the typical descriptive metadata harvested from many data providers. Once found, they need to be able to see whether the data really meets their needs which often requires detailed descriptive, contextual and provenance metadata [5]. In addition, they need to find statements about accessibility, conditions for reuse etc. Increasingly often researchers will make use of automatic workflows where the procedures need to interact with each other without intervention of humans – at least partly. In these cases very detailed metadata descriptions will be important to inform subsequent steps about the previous ones. In many cases the metadata will be widely created by the software which is also a DO in itself, i.e., clearly identified with the help of a PID and findable/operational based on rich metadata.

We can summarize some detailed metadata elements by specifying "types" of a DOs, i.e. all DOs can be distinguished by types and as for example with MIME types, operations can be associated with the types allowing researchers to broadly ignore object structure details. Assigning types systematically to all kinds of DOs and associating operations with classes of types opens the way towards automatic processing, which we *call type-triggered automatic processing*. For data science this can lead to a breakthrough towards increased efficiency, since it opens the way to systematic encapsulation as was introduced by Abstract Data Types. Data scientists would take the role of declarative working experts instead of operational ones since they now only relate data types with software types. It should be noted, however, that in data driven science there will be areas where manual workflows will remain and, in some cases, will dominate for years to come.

Metadata is mostly open and, in all cases, should be FAIR. Therefore, it will be copied, extended and reused for different purposes by researchers and services. Thus, we need to distinguish between the metadata that is part of the DO created by "authorized actors" and any other copy or extension of the metadata instance existing in the digital domain. Researchers may add "tags" to the predefined metadata which may be of great relevance for their research. However, these commentaries need to be distinguished from the "original" metadata. In various communities the term "annotations" is being used which can also be called a special type of metadata[4]. In the language community "annotations" are commentaries on the content of the DO instead on the whole DO, i.e. they are aligned with the bit sequence. This could be for example an interpretation of the tone contour of whale songs or a comment on special phenomena in parts of brain images.

Metadata is still a field with very heterogeneous approaches, as discussions in RDA have shown, which was the reason to start a series of workshops about Metadata for Machines in the GO FAIR initiative [40]. The aim is to involve specialized domain communities in the review and reconstruction of increasingly machine readable and interoperable metadata and data, or in the context of this article: interoperable DO's.

Access Permissions

For data driven science it is of crucial importance to assign rights to entities. The concept of DOs is an excellent mechanism to associate attributes defining access rights, reuse licenses, smart contracts, transaction records, etc. The already mentioned binding mechanism needs to be used to tightly and persistently bind such entities with DOs or DO classes. Appropriate operations associated with types and other attributes such as "owner" or "copyright holder" could guarantee to increase security when linked with appropriate security mechanisms. Also, the systematization of this aspect would open the door to efficient data management without losing trust, as will be required in future research infrastructures.

Finally, we should make a note about the nature of digital representations of concepts and relations as they are being used in more or less complex ontologies. In fact, digital representations of concepts and relations are DOs of specific types. This means that they should follow the same principles as specified for other DO types, i.e. they may have metadata and a PID should have been assigned to them. This does not prohibit experts exporting (for example) RDF assertions to triple stores, to look for inferences and carry out Linked Data [41] operations. Again, the use of Digital Objects merely adds stable references to facilitate this work and sustain it for the future.

We can summarize that DOs are not only technically relevant, but also scientifically significant. They help researchers

- ***to organize the increasingly complex domain of digital entities,***
- ***to associate clear and stable identifiers as anchor points of referencing in all circumstances and metadata of different kinds,***
- ***to start planning a transition towards (partial) automatic processing, and***
- ***to start assigning access permissions and reuse declarations in a stable and traceable way where this is being required.***

Finally, the systematic use of DOs will open the path towards FAIR compliant data.

[4] In some communities the concept of nanopublications comes close to these types of annotations.

5. Towards DO Based Infrastructures

The domain of data in science and industry will grow in volume and with it the heterogeneity of types and content. We can also expect that the number of tools to cope with the heterogeneity of data will increase and also the types of analysis that experts want to carry out. In section 3 we presented various approaches from IT industry to cope with heterogeneity, such as interoperability frameworks and platforms (which are usually proprietary solutions) providing an increasing number of adaptors to meet urgent needs. We argue that these can only be local and short-term solutions, and they will not help to support the paradigm shift which we are requesting. There is an urgent need to reduce overall complexity and thus to prevent severe system failures and instabilities. On the other hand, we demonstrated the power of the concept of "objects" to cope with complexity once applied systematically. In this chapter we want to look more into the technological aspects of "Digital Objects" in the domain of data. The crucial characteristics of "objects" are their capability to promote abstraction and encapsulation, to act as active entities and their fundamental role in binding closely related entities.

Definition of Digital Objects

In 2013 the Research Data Alliance was started in reaction to the huge and increasing fragmentation and inefficiency of work in the data domain. One of the first Working Groups focused on defining a core model for the organization of data, with essential contributions from R. Kahn. In 2014 RDA's Data Foundation & Terminology Group (DFT) presented its Core Data Model [42] (see figure 3) which states that a DO is represented by a bit sequence being stored, managed and served by some repositories, is referenced by a globally resolvable and persistent identifier, and is described by metadata[5]. It also states that metadata descriptions are DOs and that DOs can be aggregated in collections, which also are DOs. This definition, which was based on a broad analysis of more than 20 documented use cases from various scientific domains, only slightly differs from Kahn & Wilensky's original formulation, which stated that a DO has a structured bit sequence, a unique PID, and at least key metadata which includes the PID.

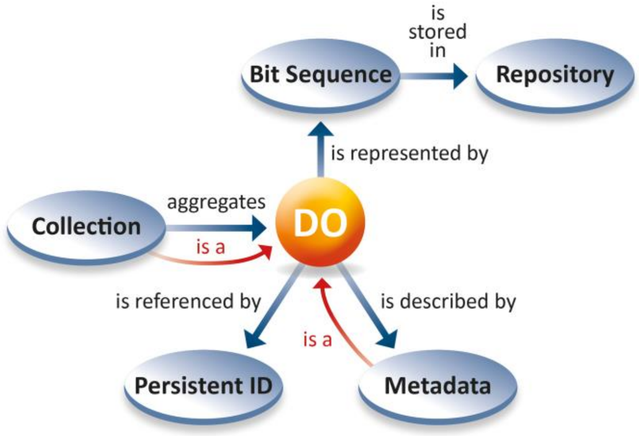

Figure 3 presents the Core Data Model centered on Digital Objects as it has been worked out in the RDA DFT group based on many use cases from various communities.

In 2014, the Lorentz Centre in Leiden organized a workshop that formulated the FAIR principles, which were subsequently published first on the FORCE11 web site for a period of community input, and then later in an article in Nature Scientific Data [43]. The principles, by their very nature as high level guiding principles, did not speak about Digital Objects, but in line with the DO concept requested that persistent identifiers and rich metadata be associated with digital entities. In the recently published "FAIR Data Action Plan: recommendations and actions from the European Commission Expert Group on FAIR data" [44] the concept of “FAIR Objects” has been introduced which we see as a synonym for FAIR Digital Objects as defined above. This emphasizes the extremely important role of persistent identifiers and Digital Objects for the future organization of our digital domain.

[5] The definition of a DO does not make statements about the content of the bit sequence, interpretation is left to other layers.

The discussions also revealed the relevance of the persistent identifier to bind closely related entities, i.e., those that are constitutional for a DO such as its membership in a collection, the locations where the bit sequences are stored and the various types of metadata (see below) which are necessary to find, access, interpret, and control reuse of the DO's bit sequence. We assume the availability of a global and persistent ID resolution system such as provided by the Handle System that resolves a PID into state information, extracted from the PID record. It is the data community at large that needs to take care that a PID resolution system is indeed persistent. Based on this assumption, it makes sense to store crucial information or pointers that allow machines to find this crucial information about the DO in this record as indicated in figure 4. In the DFT group this was called the binding role of the PIDs. An increasing number of communities see the advantage of this mechanism in organizing their data in a stable way, i.e. the PID is not just an identifier - it should include or point to all relevant information that is important to work with a DO and in that sense be resolvable to form instance meaning and/or location. To allow machines to interpret the attributes of a PID record, there is a need to associate types with them, which can be done using Data Type Registries (see below). It is necessary that repositories that make use of PID services to store this binding information specify exactly which types they are supporting.

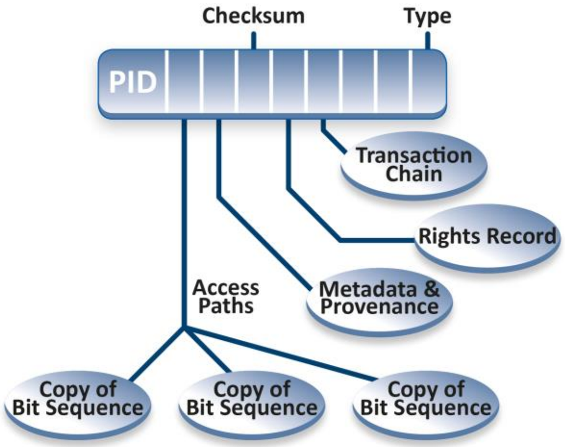

Figure 4 indicates the use of the PID record to store immediately relevant metadata and to do the binding of other entities to make DOs FAIR.

Persistent Identifier Resolution System and Digital Objects

The specifications from the RDA DFT group converge with the ideas about the core pillars of a DO Architecture from Kahn & Wilensky, which was expressed in their 2006 paper. These should consist of repositories, registries, and a global identifier resolution system resolving identifiers to meaningful state information. Some scientific communities, such as climate modeling and language research, were already widely following such a model in their data infrastructure implementations, which started around 2000. It is obvious that we are becoming increasingly dependent on a robustly functioning global PID resolution system. Therefore, it was of greatest importance to formalize the Handle System, which is the basis of more than 3000 organizational Handle services worldwide, in the international DONA Foundation, which has its location in Geneva and which is governed by an international board [45]. The Handle System consists of two layers: (1) a distributed and redundant global Handle resolution system with nodes, called Multi-Primary-Agencies (MPA), in stable centers located in almost all regions of the globe, authorized and validated by the DONA foundation to guarantee sustainability and smooth operation and (2) many Handle Service providers, which are assigned certain prefixes and which are independent in the way they serve communities. The most well-known community is the DOI community [46], which is based on a specific business model in which the registration agencies share the expenses of the central organization and the central organization guarantees the persistence of the identifiers in the event of any of the registration agencies going out of business. Another well-known community in Europe is the ePIC community [47], which has built a system with high redundancy, supported by strong computing and data centers to provide and store Handles to interested scientists and scientific communities also serving the big data sector.

RDA's Dynamic Data Citation group [48] specified 12 rules that should be followed to assign PIDs to dynamic data, which is a frequent phenomenon in science and beyond. Time stamping of each entry and associating PIDs with queries are also crucial aspects to guarantee proper referencing in the case of dynamic data. Several communities already apply these rules in the form of checklists associated with their activities.

Kernel Information Types and Digital Objects

Various activities in RDA focus on PIDs and DOs. The Kernel Information group [49], which followed up the early PID Information Type group [50], focuses on defining essential types that could be used in PID records. Different requests from various communities make it hard to define a core set that would need to be registered in accepted Data Type Registries. However, the group managed to define such a core set of 15 elements which can be used in kernel information profiles by authorized experts. The elements are taken from the PROV ontology [51], i.e., the element semantics are reused. Additional attributes may be requested in future from participants and a process needs to be worked out to allow the registration of new elements without risking a proliferation of such kernel information types. The Data Type Registry group specified a first version of a record structure of such data type registries (DTR) [52] which is now the subject of ISO standardization. DTR entries relate "types" with "actions" both identified by PIDs and "types" that can be associated with classes of DOs such as time series data in files, structured data in databases, etc., or at a finer level of granularity, such as a specific set of cells in a database. Each type can be associated with multiple functions/operations, which can be executed based on context or specific purposes. DTRs are fairly new; however, a few communities are already experimenting with the concept, since for science it would be of great value to have a declarative way of assigning functions to types. DTRs also open the way towards highly automatic workflows (see below).

Abstraction Principle and Digital Objects

In RDA it was the Data Fabric group [53] that brought various experts working on different aspects of PID- and DO-based infrastructures together, which produced a number of spinoffs. One of these was the discussion about inherent abstraction in a world of DOs as indicated by figure 5 which was presented by L. Lannom. According to this idea, users (humans and machines), are only concerned with logical representations, not the physical representations, of DOs, i.e., they know the PID, the PID record information and metadata, including type information.

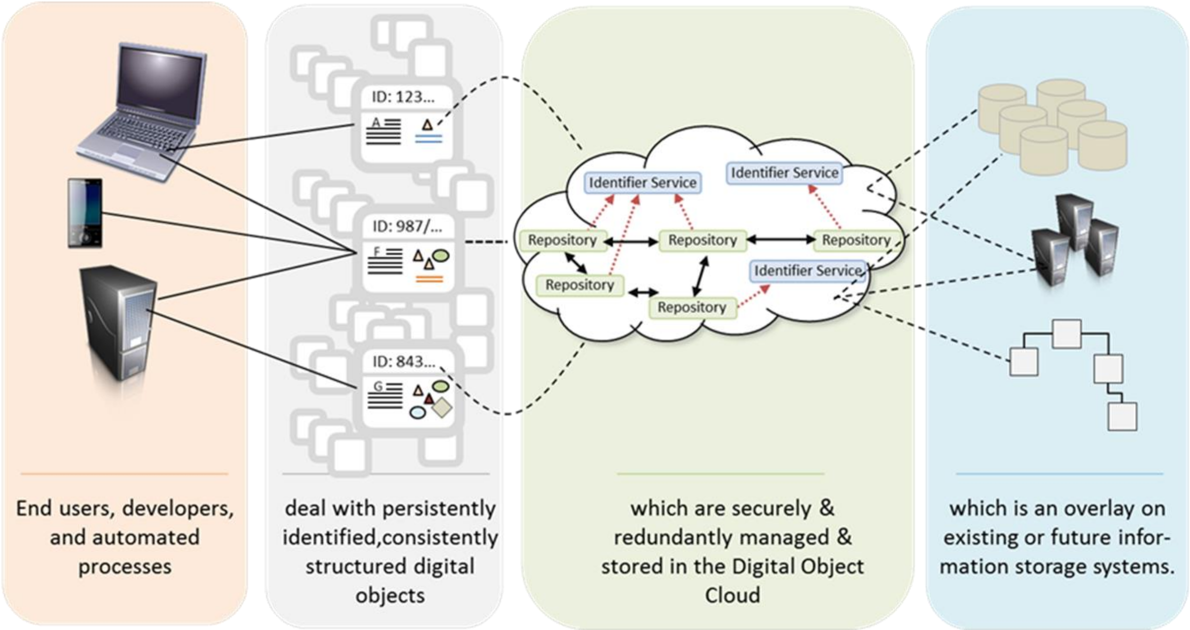
ID: 123...
ID: 987/...
ID: 843...
Identifier Service
Identifier Service
Repository
Repository
Repository
Identifier Service
Repository
Repository
End users, developers, and automated processes
deal with persistently identified,consistently structured digital objects
which are securely & redundantly managed & stored in the Digital Object Cloud
which is an overlay on existing or future infor-mation storage systems.

Figure 5 shows how the principle of virtualization can be applied to the domain of digital entities. At first instance, users are only working with logical representations of Digital Objects, i.e. they see PIDs, PID record attributes and metadata of different sorts. These are offered by repository and registry services which abstract away from the users all details about the location of the corresponding bit sequences, about their organization and modeling. For the user it is not relevant to know for example whether data and metadata are all stored in a large database, in file systems or in a cloud.

This information is sufficient to carry out a wide range of operations, perform various statistical analyses, to build virtual collections, etc. This information about DOs is generated by a cloud of repository and registry service providers. These repository and registry services build a virtualization layer on top of lower level systems, which store and process the DO implementations. The user does not need to see the details: it is not relevant for users whether data is stored in a file system, in a cloud, or in a complex database.

DOs are thus perfect vehicles to abstract away from all details, but nevertheless allow human and machine users to access all relevant entities due to the binding concept and the machine actionable kernel information types. It is also possible to 'dig down' into aggregated DO's, down to the individual assertion or concept representation level, in ever increasing layers of granularity. DOs also implement the principle that it does not matter for many operations such as "create a copy of a DO" whether the DO contains data, metadata, software, etc. However, with other operations such as the orchestration of a workflow the tools need to interpret the DO type and its metadata to support the user in making appropriate choices, for example, which DO's contained in the aggregate DO are needed for the desired operation.

Encapsulation and Digital Objects

In combination with the DTR, a strong encapsulation of DOs is achieved. Encapsulation means that the human user does not know (or care) what the internal structures are and how operations are implemented. *Strong encapsulation* means that the user does not even need to know beforehand what the operations are. The type registry contains the operations that can be executed on a DOs bit sequence. These could include not only the standard functions such as create, modify, move, delete, etc. but also analytic functions. To achieve the flexibility necessary for science with continuously new kinds of data and analysis it must be easy to integrate new operations into such a framework where easy to use tools will be crucial for acceptance. Equally crucial is that combinations of DOs (workflows and data for instance) that were used to generate a particular result are 'archived' so that the exact same versions of the software and data will be used. This would open the path towards automatic and reproducible type-driven scenarios in future.

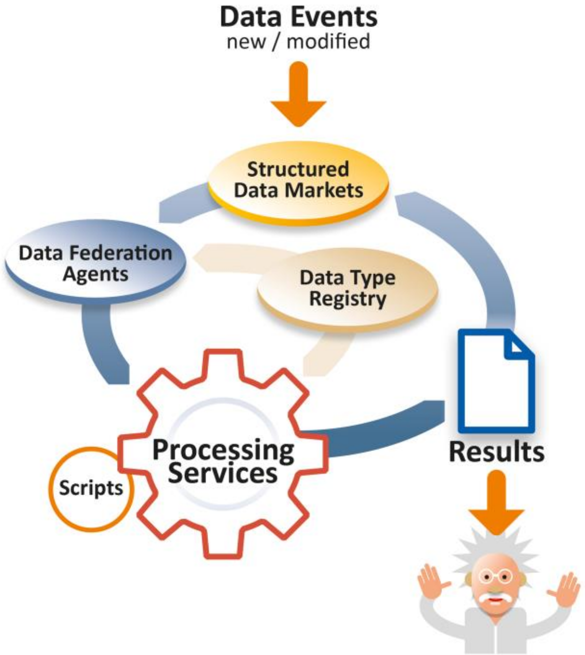

Figure 6 describes a future scenario of automatic processing of data where software agents scan a structured market of metadata offers to find useful data based on advanced profile matching and where data is processed to generate new evidences.

In RDA's Data Fabric group, a model was discussed which is schematically sketched in figure 6. New data is being produced, associated with sufficiently rich metadata and the metadata is offered via standard protocols such as OAI PMH or in future resourcesync [54] to registries which may harvest metadata from specialized labs. We call these offers of metadata about data relevant for specific use cases "*structured data markets*". Interested researchers can tune software agents to crawl the relevant parts of the structured data market for new data adhering to a specific profile, i.e., the request profile is continuously matched against the metadata of new data to check the usefulness. In case of a positive matching result the new data can be integrated into the collection to be processed and workflows can be started automatically offering new or updated results to the researcher. The role of the researcher will change from an expert who has to scan all web-sites of relevant institutions to one who specifies profiles to be matched, analyzes the results, and traces plausibility and correctness. The Data Type Registry is crucial for specifying the types to be looked for and the kind of operations to be carried out.

FAIR Principles and Digital Objects

The FAIR principles are now driving a globally accepted minimal set of behaviors that would ensure FAIRness of data and services by machines. The concept of DOs with its inherent abstraction, binding and encapsulation mechanisms is one way to lay the fundament for a FAIR compliant infrastructure. Such DOs are not just one arbitrary way to implement FAIR compliance but are themselves a system demonstrating additional behaviors necessary to achieve a paradigm shift in data management and processing. The intention behind FAIR and DOs is to initiate the momentum for such a shift. DO's will also be ideal elements to be automatically measured for their degree of FAIRness.

DOs *to be Findable*: The DO concept requires the association of PIDs and (rich) metadata with bit sequences representing some content and ensures that DOs can either be found by using the PID or by using the exposed metadata. Due to the binding function both paths give humans and machines access to all entities of the DO.
DOs to be Accessible: The DO concept enables an infrastructure that makes data and metadata accessible and it supports all requirements with respect to open and fee-to-use protocols and authentication and authorization.
DOs to be Interoperable: The DO concept takes care of interoperability at the basis of data organization (all concepts in the data are represented by dereferencable DOs themselves, namely the PID of the concepts used in the data), which would reduce ambiguities and inefficiencies considerably and it facilitates structural and semantic interoperability due to providing stable references and dynamic mappings between PID systems.
DOs to be reusable: The DO concept facilitates reusability due to the binding of all relevant entities, which would help for example humans and machines to find the corresponding blockchain entry that could contain smart contracts (machine actionable licenses) and transaction information in a safe way and provenance records.

In some cases, DOs directly implement the FAIR principles; in some others they facilitate the implementation of the FAIR principles.

Metadata and Digital Objects

In general, models including the RDA DFT model described above simply refer to "metadata" without going into detail although metadata is crucial, as the FAIR principles clearly endorse. There are two aspects that make it so difficult to structure the metadata domain and to come to harmonization:

- Metadata statements in their essence are assertions about Digital Objects (which can be anything digitally represented) and are part of the DO to make it findable, accessible, interoperable and re-usable.
- Metadata statements are made by different domain actors (humans, machines) with different roles and perspectives for different purposes at different times.

There are quite a number of different types of metadata assertions about the DO such as type information to facilitate automatic processing, descriptive information to facilitate searches, scientific collection building and inferencing, description of scientific information which usually goes much deeper than the usual descriptive information, in order to enable deep scientific analysis, system/state information to help managing DOs, context and provenance information to cover the relevant aspects of the creation process, access permission and license information and information about transactions. Many of these types are not yet strictly defined and agreed upon between and within the various communities as evaluations in RDA and elsewhere have shown. Some introduced new types of categorizations such as "intrinsic" and "user" metadata [55], yet it needs to be shown whether this helps in structuring the metadata domain. Since metadata is so crucial there is an urgent need for convergence on definitions.

There is a wide agreement that schemas and semantic categories should be FAIR as well, i.e., be defined in open registries, have PIDs and have some metadata which is part of the definition. The RDA Metadata Directory group [56] is offering a place to register schemas so that reuse and machine readability is guaranteed and also schema.org [57] is a place to store a large number of schemas that are in use mainly to tag web-pages, but can also be extended to expose indexable subsets of terms in the metadata of for instance FAIR data points. For that purpose, metadata will also be exported as html versions to be read by global search engines such as Google and as (augmented) RDF triples to enable their integration in Linked Open Data and to foster semantic analysis and inferencing. In this respect, we should refer to the work on Knowlets as depicted in figure 7 which represent a way to point to core concepts and to represent their semantic relationships in a specific semantic space and thus extract relevance, unforeseen relationships, etc. Knowlets are highly dynamic constructions due to continuous updates and these changes lead to management challenges, i.e. Knowlets need to be seen as DOs that are subject of frequent versioning and probably require provenance tracking. Knowlets, originally developed to deal with conceptual overlap between non-co-occurring concepts in the concept space, can also be used to represent assertions about any other DO than a representation of one particular concept. They could represent any other DO as the collection of all assertions about that DO, each with its own provenance. This is one way to deal with near-sameness in the concept space of DO’s. The reasoning behind this approach is further elaborated and explained in reference [58].

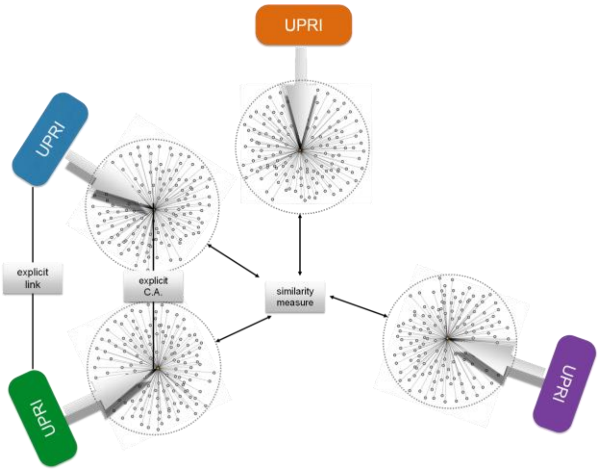

Figure 7: Knowlets are clusters of machine-readable assertions about a core concept (DO) in the center. Machines can calculate the conceptual overlap between knowlets for many different perspectives. Knowlets can be filtered on the types of the subject, predicates and objects in the Knowlet and thus machines can group DO’s based on their types and metadata based on a dynamic matching service. Detailed description in [ref 58].

Some metadata such as descriptive and scientific metadata will be open to inform others about available data and should be made available via open protocols such as OAI PMH and Resourcesync. Openness implies reuse for various purposes in different contexts and thus will lead to modifications and enrichments. These new metadata variants need to be separated from the original metadata object which is associated with the originating DO, which should only be changed by authorized experts. Other metadata such as access permissions and transactions will not be open and instead need to be highly protected. For special kinds of metadata such as access permissions, smart contracts and transaction records special schemas will be required which, however, also need some standardization.

Recently there is a trend towards introducing modularization in the metadata domain to break down complexity. The CLARIN research community developed a flexible metadata system based on components [59] which are XML snippets that can be stored and combined by users and which make use of registered metadata categories. In doing so, they allow users to describe their DOs in their own way while still guaranteeing interoperability by using only agreed and registered categories. The RDA Metadata groups [60] are working on packages with similar intentions and the GO FAIR community specified the need for "Atomic Metadata Templates" [61]. It seems at least that this kind of modularization based on proper agreements can help overcoming the heterogeneity, but it will not be easy to make progress due to the sociological challenges. What we need is a highly flexible system where assertions about conceptual mapping and DO's in general can be made by anyone in the community with provenance and 'authorship' of the assertion, so that for example someone can 'assert' that [PID1] in vocabulary [x] is referring to the same concept as [PID2] in vocabulary [y] and in fact, at a higher level of abstraction, anyone can make assertions about the relationship(s) between any pair of DO's in the 'Internet of FAIR Data and Services'.

Automatic Workflows and Digital Objects

In science, automatic workflows are currently not widely used. The experts often rely on ad-hoc scripting and manual work since these allow them to react on the specific needs of an analysis. However, the increasing availability of experts who can create flexible and parametrized workflows will grow and new tools such as for example Jupyter notebooks [62] will make it easier to experiment with and reuse workflows. The amount of re-useful data and sustainable, well documented and versioned workflows will increase exponentially and despite all thresholds still to overcome there will be increasingly rich metadata including detailed typing that will be the basis for successful profile matching as indicated above. It seems to be impossible in the future to continue our current attitude where we still use personal interactions to find useful data. But we should not underestimate that the way towards a scenario as sketched in figure 6 will take time.

Nevertheless, some labs are making use of limited workflow tools for part of their tasks. These tools range from web-services such as Weblicht [63] where technically inexperienced users can do their orchestration for NLP (Natural Language Processing) tasks allowing them to carry out, for example, named-entity detection in texts to tools that allow experienced users to formulate their workflows using new types of tools such as Jupyter notebooks, or in a typical workflow language such as Common Workflow Language [64], and to store and share their workflows with others using frameworks such as myExperiment [65]. Increasingly important for such frameworks is the use of containers to transport DOs including their contexts. DOs are spanning a graph when all related entities that are being considered require some form of serialization to carry out the transfer. Research Objects [66] address exactly this point: how to serialize complex DOs for transfer within workflows and package this into containers.

Security and Digital Objects

Despite the general agreement that open data should be the default, especially for publicly funded research outputs, much data will be protected for various reasons (privacy, licenses, economics, etc.). Security of data and control of transactions are thus important for maintaining trust. In this paper we will not elaborate in depth on security issues. However, we claim that DOs are an excellent basis on which to define security measures. Since each DO has a PID and is associated with some fingerprint information (checksum, hash, etc.) which should be anchored in the PID record, identity can always be proved. The binding concept introduced above allows associating pointers to an authorized access permission record which is important when different copies of the DOs' bit sequences are being stored in different repositories and to a blockchain entry that stores smart contracts which include license information in machine actionable ways. This clearly referenceable and closely bound blockchain entry can be used to store all relevant events such as transactions of the specific bit sequence in a way that cannot be changed. It has been shown that blockchain technology as it is known today can only be used to store small amounts of data such as formal description of events. At all times the owner or copyright holder can thus check what happened with their data even in cases where an automatically driven data market scenario would be in place. Once bit sequences have been copied to other domains in conformity with contracts, it will be hard to trace where this data is being used. Criminal intent would allow using data in ways not conforming to contractual statements. Other more complex mechanisms would have to be used to prevent such misuses.

Interfaces and Digital Objects

In the previous paragraphs we claim that DOs form an optimal way to organize the data domain by their binding, abstraction and encapsulation characteristics, which can be seen as a step towards the needed paradigm shift. Repositories and registries can be built according to these principles, as some domains are already doing, i.e., the PID record is used to bind relevant entities and machines can easily find all parts. The remaining question is whether we need a new way of interfacing with DOs as sketched in figure 8. In this figure the term Digital Object Interface Protocol[6] is being used to describe a domain where all participants offer DO based interfaces independent of the type of data organization they have chosen internally. Adapters will be needed to map between the data organization as requested by the DO concept and those that are being used by repositories.

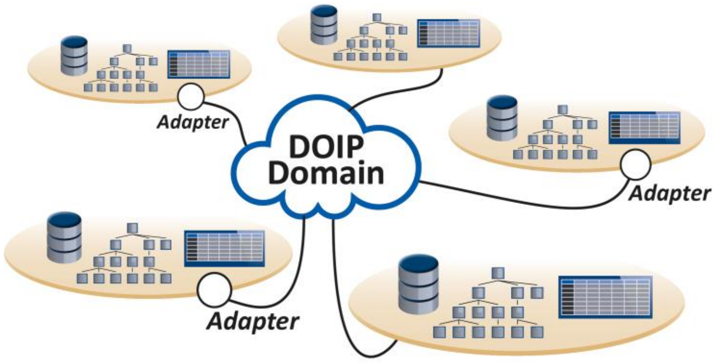

Figure 8 indicates the huge potential of the DO Interface Protocol creating a domain of interoperability between repositories all using different methods of organizing and modeling their data. Some repositories may already have an organization that directly maps to DOIP where adaption is trivial.

Figure 8 shows some similarity with the role of TCP/IP for the Internet which can be indicated by a simple hourglass analogy. There is no doubt that TCP/IP had a structuring and unifying effect. Only a wide spread acceptance of a simple, open and specified DOIP protocol will lead to wide application of the basic DO model which serves to come to explicitly defined relations between all relevant entities and thus to an implementation of the FAIR principles and to machine actionability. When, for example, a DO is being moved to a new repository with completely different internal data organization, we need to make sure that the DO is fully transferred and accessible via the same basic mechanisms as defined by the DFT core model. Such transfers include serialization mechanisms and, as always, we need to distinguish between small and big “bit sequences” to be transferred. For DOs encapsulating big data one will have separate mechanisms to transfer the bit sequences.

The first question to address is whether existing interface protocols would be suitable to support the domain of DOs. In this paper we limit ourselves to looking at the SOAP and REST standards. In the late 90s web services became a very important issue of concern to foster distributed processing. Servers offering useful services should be invoked by other services in the Internet and protocols were required to support the exchange of structured information. In 2003, SOAP V1.2 was published as an official W3C recommendation as the XML based messaging protocol specification within a stack of components to support web services which could exchange basically all kinds of different message types. For general transport, SMTP and HTTP were chosen to pass firewalls, although binding to other protocol types was possible. Thus, SOAP could be used to exchange "Digital Objects", however, they would have to be serialized and embedded in an XML "container". It would be the task of the services on both sides to discover the special organization of Digital Objects as described by the RDA DFT Core Model, i.e. SOAP could be used as a layer to support the exchange of Digital Objects, but SOAP itself does not have any knowledge of what a DO is.

[6] It seems that we will finally agree on the term DO Interface Protocol instead of DO Access Protocol. We see both as synonyms in this paper.

In 2000, R. Fielding presented his REST style specification [67] enabling RESTful web services relying fully on the HTTP protocol and using the HTTP commands (GET, OUT, POST, DELETE), i.e., commands that were typically designed to access resources are being used to exchange packets of information between web services. Unlike SOAP, REST does not require XML embedding, but can exchange various formats (CSV, RSS, JSON, etc.) which makes the creation and processing of messages often much simpler and faster. This is why REST is often preferred over SOAP, and in many applications even SOAP makes use of the REST protocol.

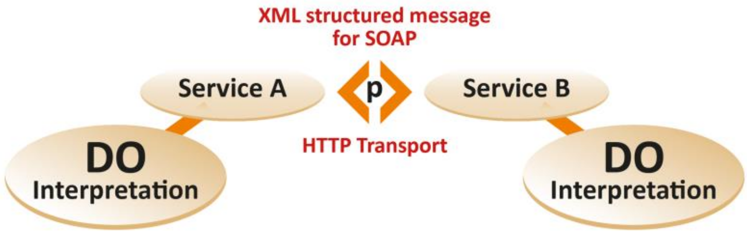

Figure 9 indicates the possible usage of SOAP or REST in the context of the exchange of DOs. They could only serve as protocol to transport serialized bit sequences that are part of the DO. Some may argue that directly TLS could be used, however, there is already much software out there using REST.

In CNRI, DONA, RDA-DF, GEDE and C2CAMP, the experts are discussing the nature of a Digital Object Interface Protocol and recently DONA released a second version of that protocol as a core element of the DO Architecture. It assumes that a client wants to access a Digital Object whatever the context and the lower transport layers are and that the client wants to execute some operations on the bit sequence of the DO. Arguing from the view point of data organization the client must be able to access all entities that belong to the DO. Assuming the availability of the PID of a DO, the client must be able to access DO's type, its various metadata types, and the possible operations. In addition, it must be possible to invoke these operations. This description also implies that at the DOIP level the DO's content represented by its bit sequence will not be touched, i.e. format conversions or any other kinds of operations that will manipulate the content must be carried out by operations that are being invoked by the client. There might be a set of standard operations which are true for all DOs such as the typical management operations create, delete, retrieve, update etc. but it has to be exactly defined what these operations will do. A create operation for example needs to not only store the bit sequence in some repository but also create a PID, create and store the various metadata types using the defined structures, invoke the services that offer descriptive metadata for harvesting etc., but also store all relevant relations in the PID record.

client
p
DO
Digital Object
Interface Protocol

Figure 10 describes a typical client-server interaction with actors on both sides which have knowledge about what a DO is. At the right side it could be typically a repository with some services which can transform DOIP based interactions to the specific solution of data organization and modeling.

The ITU X.1255 "Framework for discovery of identity management information" introduces a Digital Entity Interface Protocol[7] at an abstract level. It states *"The digital entity interface protocol defines the method by which the entity communicates with a repository for the purpose of invoking operations on the digital entities for which the repository provides access. These operations can be used, in particular, to access specific metadata records by their identifiers; but such records can also be accessed semantically through other means such as dedicated registry "apps" and web browsers."* In addition, it specifies that any operation on digital entities includes a few elements all identified by a PID:

- *EntityID: the identifier of the digital entity requesting invocation of the operation;*
- *TargetEntityID: the identifier of the digital entity to be operated upon;*

[7] X.1255 uses the term "entity" instead of "object".

- *OperationID: the identifier that specifies the operation to be performed;*
- *Input: a sequence of bits containing the input to the operation, including any parameters, content or other information; and*
- *Output: a sequence of bits containing the output of the operation, including any content or other information.*

Other important aspects for the design and acceptance of a new type of DO-related protocol will be the supported security aspects and its integration into the infrastructure landscape in the scientific domains which evolved during the last decades which involved large amounts of efforts and funding. For interfacing, most infrastructures are using HTTP and REST based methods. It needs to be worked out how a DOIP can be mapped to the existing designs to come to the adaptors mentioned in figure 8. It should be noted that the recently released version 2.0 of the DOIP [68] states that "*DOIP can be tunneled through any secure communications protocol and the DOIP itself can be used to determine the choice of such protocol.*"

6. Conclusions

In the domain of digital data we are facing a paradox which C. Borgman described at her RDA plenary talk in 2014 in Amsterdam as *"data, data, everywhere, nor any drop to drink*" in paraphrasing S. T. Coleridge who used the word "water" instead of "data". What she expressed is the fundamental challenge which we are faced with: on the one hand generating continually increasing amounts of data with increasing inherent complexity and on the other hand having little in the way of ideas and approaches to optimally and efficiently use this wealth of data for societies and economies. Data reuse seems to be confined to a small and only slowly growing 'digital elite' and reduced to the very small top of an increasingly large iceberg. This is also true in industry where the normalization of much data to well-structured data warehouses seems to have to be replaced by a new and much more loosely specified paradigm called "data-lake" which clearly indicates the dilemma we are in and how we are at a loss when it comes to globally agreed, scalable and open solutions.

We can identify four fundamental challenges to tackle this paradox in the coming decades:

- Understanding how to extract knowledge from this lake of data and formalizing this knowledge suitably.
- Incentivizing the research community to increasingly publish data and services as DOs that are FAIR at the source.
- Integrating and analyzing this knowledge which has been extracted from different virtual collections to the benefit of societies and economies.
- Defining and implementing a stable fundament for all these actions that holds for decades if not centuries so as not to end up in a "data tower of Babylon".

The last challenge is continuously underrated in the current discussions although several surveys clearly indicate that already now a large percentage of the current inefficiencies in data practices are due to bad data organization and bad quality at all levels. This demands global alignment of policies and infrastructure building which is primarily not a scientific endeavor, and which is associated with risks for industry. Initiatives like the European Open Science Cloud, the USA Data Commons and the African Open Science Platform should be aligned to collectively drive the convergence towards globally approved, minimal standards to ensure FAIRness and cross disciplinary interoperability.

Therefore, in their recent paper, Wittenburg & Strawn focus on this infrastructure aspect and argue strongly for convergence to simple standards for the data domain, the approach that was so important in earlier infrastructure building to open a phase of massive exploitation creating wealth and jobs. It is also obvious that there is a global trend to start initiatives working hard on finding agreement on such standards and that there is an increasing pressure from funders and in industry to work out suggestions that would overcome the current state of destructive fragmentation.

The first promising steps towards convergence have been made by CODATA at policy level, by RDA suggesting improvements on many detailed specifications and through the FAIR principles by many groups of experts now collaborating in the GO FAIR initiative to work on FAIR compliant implementations. But despite all efforts we lack a formal suggestion for a minimal standard that could lead to a groundbreaking change. In 2013 The RDA Data Foundation & Terminology group, which was later followed up by the Data Fabric group, defined its Core Model based on a broad analysis of use cases and started referring to papers on Digital Objects from Kahn & Wilensky, after seeing the crucial synergy between the concepts. In parallel, CNRI invested serious efforts to establish a global Handle Resolution system based on a sustainable business model and guided by an independent Swiss foundation and intensified their work on specifying the Digital Object Interface Protocol, both being essential for a functioning domain of DOs.

In this paper, we have shown that the basic concepts behind “digital objects”, i.e., abstraction and encapsulation, were crucial to build complex software systems, were at the source of new storage solutions such as cloud storage, and that the DO concept is not just an IT concept, but that it can help structuring the complex domain of digital entities in science. Industry is working on a range of different solutions tackling heterogeneity, but at least with respect to the “reference architectures” as they are worked out in production industry, we can see complementarity. The various dimensions described in section 5 above indicate why we believe that by systematically applying the DO concept for building data infrastructures we can indeed build the stable fundament which will be required to overcome the paradox described by C. Borgman. It will help to create an interoperable domain for global data management and analysis on top of which investments on appropriate data stewardship focusing more on the content of data as demanded by the FAIR principles and extended investments in knowledge extraction and integration will make sense.

The short comparison between the widely agreed FAIR principles and the concept of DOs indicate that in some cases DOs directly implement the FAIR principles and, in some others, they facilitate the implementation of the principles. The concept of DOs will not address all interoperability problems such as for example caused by semantic heterogeneity, for which approaches such as Knowlets may be needed, but it should be emphasized again that such dynamic, machine readable (meta)data entities are (and should be) DOs by themselves. So we can argue that DOs of many different types, including those addressing semantic heterogeneity, semantic drift and near sameness are a basic element of the core solutions to the current data infrastructure problems. They represent a means to overcome the organizational and referential issues that are causing so much inefficiency and are preventing automatic procedures from being applied more frequently.

It is very promising that there is an increasing convergence between the concepts being discussed in a number of these globally active initiatives such as CODATA, RDA, GO FAIR and DONA, but the final convergence will only happen when we concur on minimal, globally accepted standards with maximum freedom to operate in innovative solutions, comparable to what happened in many explosively useful infrastructures. The acceptance of such minimal standards at the policy level by funders, publishers, repositories and science policy makers will drive the convergence that is so badly needed to make the transition to open and data driven, machine-assisted science.

We therefore call on all these parties to collectively contribute to the convergence of an ‘Internet of FAIR Digital Objects’.

Acknowledgements

We would like to thank the C2CAMP and GEDE colleagues for their many valuable comments and here in particular our colleagues Larry Lannom and Dimitris Koureas.

References

[1] Peter Wittenburg,George Strawn: Common Patterns in Revolutionary Infrastructures and Data; http://doi.org/10.23728/b2share.4e8ac36c0dd343da81fd9e83e72805a0

[2] Bitkom: Big Data & AI Summit; https://www.big-data.ai/

[3] Peter Wittenburg et. Al.: Report Industry Side Meeting at the 11th RDA plenary in Berlin; https://www.rda-deutschland.de/intern/dateien/20180319-20_rda_p11_industry-side-event-report-final.pdf

[4] Vint Cerf: Google's Vint Cerf warns of 'digital Dark Age'; https://www.bbc.com/news/science-environment-31450389

[5] Mark Wilkinson et al, The FAIR Guiding Principles for scientific data management and stewardship, https://www.nature.com/articles/sdata201618

[6] RDA¸ http://rd-alliance.org

[7] Larry Lannom: Managing Digital Objects in an Expanding Science Ecosystem; https://www.rd-alliance.org/sites/default/files/CENDI-15.Nov_.17-Lannom-Final-2.pdf

[8] GO FAIR; https://www.go-fair.org/

[9] Term “Object”; https://en.wikipedia.org/wiki/Object_(philosophy)#Etymology

[10] Subject-Object Split; https://www.wissen.de/lexikon/objekt-philosophie

[11] Term "Property"; https://en.wikipedia.org/wiki/Property

[12] R. Kahn, R. Wilensky: A Framework for Distributed Digital Object Services, 1995; http://www.cnri.reston.va.us/home/cstr/arch/k-w.html

[13] R. Kahn, R. Wilensky: A framework for distributed digital object services, 2006; https://www.doi.org/topics/2006_05_02_Kahn_Framework.pdf

[14] Cross-Industry Working Team: Managing Access to Digital Information; https://www.cnri.reston.va.us/xiwt_archive/ManagAccess.pdf

[15] P. Wittenburg, M. Hellström, et. al.: Persistent identifiers: Consolidated assertions. Status of November, 2017; https://zenodo.org/record/1116189#.W9VsNDGNzb0

[16] A. Kay: Smalltalk; https://en.wikipedia.org/wiki/Alan_Kay

[17] Object Oriented Programming; https://de.wikipedia.org/wiki/Objektorientierte_Programmierung

[18] B. Liskov, S. N. Zilles: Porgramming with Abstract Data Types; http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.136.3043&rep=rep1&type=pdf

[19] Abstract Data Types; https://en.wikipedia.org/wiki/Abstract_data_type

[20] Objet Storage; https://en.wikipedia.org/wiki/Object_storage

[21] Fedora Commons; https://duraspace.org/fedora/

[22] DSpace; https://en.wikipedia.org/wiki/DSpace

[23] J. Hendricks, et. al: Improving small file performance in object-based storage; http://www.pdl.cmu.edu/PDL-FTP/Storage/CMU-PDL-06-104.pdf

[24] Ian Foster & Kesselmann: The Grid: Blueprint for a new computing infrastructure; https://dl.acm.org/citation.cfm?id=289914

[25] Amazon S3; https://en.wikipedia.org/wiki/Amazon_S3

[26] Data Warehouse; https://en.wikipedia.org/wiki/Data_warehouse

[27] Data Lake; https://en.wikipedia.org/wiki/Data_lake

[28] C. Gunden: Why It Matters; https://blog.nucleusanalytics.com/data-warehouse-vs.-data-lake-and-why-it-matters

[29] NoSQL Databases; https://en.wikipedia.org/wiki/NoSQL

[30] IBM Watson; https://www.ibm.com/watson/

[31] Peter Wittenburg et. Al.: Report Industry Side Meeting at the 11th RDA plenary in Berlin; https://www.rda-deutschland.de/intern/dateien/20180319-20_rda_p11_industry-side-event-report-final.pdf
[32] RAMI4.0; https://ec.europa.eu/futurium/en/system/files/ged/a2-schweichhart-reference_architectural_model_industrie_4.0_rami_4.0.pdf
[33] Industrial Internet Consortium; https://www.iiconsortium.org/
[34] Industrial Data Space; https://www.fraunhofer.de/en/research/lighthouse-projects-fraunhofer-initiatives/industrial-data-space.html
[35] Industry 4.0; https://en.wikipedia.org/wiki/Industry_4.0
[36] DOBES; http://dobes.mpi.nl/
[37] ENES; https://portal.enes.org/data/enes-model-data/cmip5/resolution
[38] F. J. Kurfess, et. al.: Knowlets: Components for Knowledge Management; http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.25.3961
[39] B. Mons: FAIR Science for Social Machines: Let's Share Metadata Knowlets in the Internet of FAIR Data and Services; http://www.data-intelligence-journal.org/static/publish/B5/D1/28/A85A05492CA4E681827A4F8BF7/Barend_Mons.pdf
[40] GO FAIR Metadata for Machines; https://www.go-fair.org/events/m4m-workshop/ & https://www.go-fair.org/resources/go-fair-workshop-series/metadata-for-machines-workshops/
[41] Linked Data¸ https://en.wikipedia.org/wiki/Linked_data
[42] RDA DFT Core Terms and Model; http://hdl.handle.net/11304/5d760a3e-991d-11e5-9bb4-2b0aad496318
[43] M. D. Wilkinson, et. al.: The FAIR Guiding Principles for scientific data management and stewardship; https://www.nature.com/articles/sdata201618
[44] FAIR Implementation Report: https://doi.org/10.2777/1524
[45] DONA Foundation; https://www.dona.net/
[46] DOI Foundation; https://www.doi.org/
[47] ePIC Consortium; https://www.pidconsortium.eu/
[48] RDA Dynamic Data Citation; https://rd-alliance.org/group/data-citation-wg/outcomes/data-citation-recommendation.html
[49] RDA Kernel Information; https://www.rd-alliance.org/groups/pid-kernel-information-wg
[50] RDA PID Information Types; https://www.rd-alliance.org/groups/pid-information-types-wg.html
[51] W3C PROV; https://www.w3.org/TR/prov-overview/
[52] RDA Data Type Registry; https://www.rd-alliance.org/group/data-type-registries-wg/post/data-type-registry-first-prototype.html
[53] RDA Data Fabric; https://www.rd-alliance.org/group/data-fabric-ig.html
[54] OAI ResourceSync; http://www.openarchives.org/rs/1.1/resourcesync
[55] CEDAR; https://metadatacenter.org/
[56] RDA Metadata Standards Directory; https://www.rd-alliance.org/metadata-standards-directory
[57] Schema.org; https://schema.org/
[58] Barend Mons: FAIR science for social machines: http://www.data-intelligence-journal.org/p/10/1/
[59] CLARIN Component Metadata; https://www.clarin.eu/content/component-metadata
[60] RDA Metadata IG; https://rd-alliance.org/groups/metadata-ig.html
[61] GO FAIR Atomic Metadata Templates; https://osf.io/qe9fa/
[62] Jupyter Notebook; http://jupyter.org/
[63] CLARIN Weblicht Workflow; https://weblicht.sfs.uni-tuebingen.de/weblichtwiki/index.php/Main_Page
[64] Common Workflow Language; https://www.commonwl.org/
[65] myexperiment; https://www.myexperiment.org/home
[66] Research Objects; http://www.researchobject.org/
[67] R. T. Fielding: Representational State Transfer (REST); https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.html
[68] DOIP V2.0 : https://www.dona.net/sites/default/files/2018-11/DOIPv2Spec_1.pdf