RDA Data Foundation and Terminology DFT 3: Snapshot of DFT Core Terms

Gary Berg-Cross, Keith Jeffery, Bob Kahn, Larry Lannom, Raphael Ritz, Herman Stehouwer, Peter Wittenburg, Thomas Zastrow, Zhu Yunqiang

State: July 2015

Version 1.5

The general outline of documents from DFT WG is as follows:

- DFT 1: Overview
- DFT 2: Analysis & Synthesis
- **DFT 3: Term Snapshot**
- DFT 4: Use Cases
- DFT 5: Term Tool Description

1. Pre-remarks

1.1 Introduction to the Snapshot

The Data Foundations and Terminology (DFT) working group of the Research Data Alliance (RDA) understands that even the definitions of the core terms[1], as described in this document should be seen as snapshots in the data management and infrastructure field that is highly dynamic. Therefore this note should be viewed as capturing a moment in time – one in which we try to consolidate and organize ongoing discussions about term concepts[2]. The vision is to use such a snapshot as a platform to accelerate discussions towards real, working agreements on terminology within RDA and across the worldwide data community.

1.2 Scope

The task of RDA is to improve conditions for data sharing and re-use. In order to do this, the RDA must focus on the domain of registered, visible and accessible digital data[3], knowing that current data management and access practices are not adequate to the challenges which include the observations that:

- much data are not visible since there is inadequate metadata (descriptive, contextual and provenance etc.[4]) documentation,
- much data exist in local stores in some undefined state and do not have an external, referencable "identity" that would enable unique access, checks, etc. and

[1] Appendix B includes terms that have been in discussion, but those are not as essential or developed for our data organization discussions as the selected core terms. We want to mention them in this document to maintain an overview and we simply give indications of what has been discussed and may be further developed by relevant RDA WGs and/or follow-up work on RDA terminology.

[2] A larger set of terms and definitions is viewable in the online DFT term tool at http://smw-rda.esc.rzg.mpg.de/index.php/Special:AllPages

[3] Visibility and accessibility is meant for humans and machines.

[4] Also provenance information is one form of metadata, but it is important to mention the various packages.

- much data are not accessible in an easy way.

We acknowledge that there is much data around in non-digital formats (for example on paper or on analog tapes) and on private (dark) storage systems that do not yet fulfill the requirements of being digital entities that can become seamlessly part of the sharable and re-usable data domain. We also acknowledge that much data are not properly described, registered or documented and still exchanged by traditional mechanisms (exchanging some carriers such as tapes etc. , sending email attachments, etc.) which do not support, for example, any easy form of provenance tracking.

Stepwise and incremental efforts to improve data sharing and re-use by more consistently employing extant standards remains in play. However, in the view of RDA's Data Foundations and Terminology (DFT) working group, accelerated improvements can be made by better standards and reference models. Providing an improved reference schema for a data organization and its resulting metadata would stimulate a direction for data professionals and software builders. Such a schema could represent one direction that data management can move to that will help us keep up with, if not master, the challenges of data deluge. It is worth noting that as Kenneth Haase[5] has observed quality metadata's value follows something like Metcalfe's law for the economics of network technologies. That is:

> "the value of metadata rises as the product of the log of the corpus size and the log of the size of the user community"

For example, good metadata was less an issue with small DBs (an 100 attributes and a 1000 instances) used by a handful of researchers in particular communities. These produce what has been called dark data. For these simple, ad hoc organizational schemes, based on dimensions like time, location, place, one type of sensor, keywords known to the community etc., provide enough contexts that personal knowledge fills in to allow effective retrieval and identification of relevant data. In a Big Data era with much more data to be shared across different communities such informal efforts are not adequate. This means that an issue such as the lack of proper semantics to support automated processing of metadata to assist with data finding, access, integration and sharing become increasingly important.

1.3 Recent Observations

A recent study[6] carried out during the last two years and including about 50 interviews with data professionals from departments/institutes from different disciplines and intensive discussions with data professionals from different disciplines in about 75 meetings, supported in general our impressions of the current state of data. These interactions did not focus so much on data publication issues, which are an important part of the data life cycle and a hot issue for scientists, but instead focused mainly on the scientific data processing (management, analysis, etc.) phase.

The main conclusions addressing these issues of the core of the scientific data processing work which we can draw from this report are:

- Data Management (DM) and Data Processing (DP) are too time consuming and costly due to the heterogeneity of data organization, in particular regarding logical information[7] and how

[5] Haase, Kenneth. "Context for semantic metadata." *Proceedings of the 12th annual ACM international conference on Multimedia*. ACM, 2004.

[6] Published on the RDA Europe site: http://europe.rd-alliance.org

[7] With logical information we summarize different kinds of metadata information associated with DOs such as contextual and provenance metadata, PIDs, relations, rights, etc. The Metadata and Provenance IGs need to work on the details of some of this meta-information.

it is documented. People see the need to change habits and routines, but have not agreed on policies and best practices needed to change them.
    - Consider that, for example, one of the key biologists in a large research institute is spending 75% of his time on manual data management, we can assume that too much money and human capital is wasted on current data management procedures.
- Federating and networking data including logical layer information which is relevant for
    - provenance tracking,
    - understanding data creation in context,
    - checking identity and integrity, etc.

  is so costly that in practice it is not done in most research labs, although most data professionals understand that in the long run, they must change their practices.
- DM and DP are not prepared to deal with "Big Data" because labs lack or inadequately use automated procedures incorporating proper data organization mechanisms[8].
- Due to a lack of software that is supporting proper data organizations, researchers continue to create legacy data that cannot be easily integrated into the growing data domain.
- When we look at these practices and their implications, it is obvious that we need to change and that the DFT working group can play a critical role in contributing towards a solution that promotes focused discussion about data organizations with an intention to indicate solution directions.
- We also need to develop and start on staff training to inform our young people how we can overcome this situation.

It also needs to be noted that whatever changes are adopted for dealing with data, it will only work when efficient software is used to support these new features. We must avoid putting more data administration burden on the researchers, e. g. when registering data, associating a PID including relevant properties such as names and checksums and other metadata. The bulk of this needs to be done automatically by the software almost transparently to the researcher. This would support the scenario that a researcher who wishes to share and re-use data only needs to know where the PID can be found, how it is used for referencing and can then copy & paste it, for example, into publications.

We note that a number of relevant components for moving towards an improved situation are being addressed by RDA WGs. For example the Data Description Registry Interoperability WG addresses the enhanced role that machine-to-machine readable interfaces of data registries and repositories have in areas such as descriptive metadata:

> *".. the issue of wider discovery is often addressed either by metadata aggregation (collecting records from these registries and making them accessible through an aggregator portal) or federated search (exchanging queries and search results within a federation). However, the main problem is providing scientists search results for datasets that are actually relevant to their research. Such relevance depends on research context, and as a result enabling cross-platform discovery includes providing a connected graph of researchers, research activities (projects and grants), research datasets, publications and other research outcomes and research concepts" From the RDA WG site-*
> *https://rd-alliance.org/group/data-description-registry-interoperability.html*

[8] As noted before this is partially due to a lack of registered syntax and formal semantics about data. Even what seems like"good" metadata documentation about data (such as a subject area name) is only understandable to humans and not processible by automated systems. Efforts underway as in RDA WGs such brokering governance or semantic interoperability may help here.

Likewise in another scenario intelligent brokers (also under discussion as part of an RDA Broker IG) may provide help finding relevant data by properties some of which are captured along with the data.

In Appendix A we include a few examples to indicate typical current practices and their draws back in terms of proper data handling.

1.4. Term Framework

While the interactions that were included in the recent, previously mentioned study were conducted to better understand today's data practices, the models collected by the DFT group describe steps or approaches that have been taken to overcome some of the inefficiencies. The conclusions from the interactions complement our DFT model overview and analysis papers covering 22 models that form the basis of our conceptualization.

Almost all terms the DFT group has looked at are not new. They are being used in other communities for some time identifying different aspects within a concept domain. The RDA needs to respect the definitions already in use, but as a first step needs to work out its own conceptual domain and in particular the relations between these concepts to be used in discussion and in successfully finding solutions that may help us to overcome the many barriers. A next step is to understand the differences between current use and the RDA target with a justification for a change from existing usages. Finally, the earlier term definitions including the model concepts behind them are rolled out, the earlier community conversations are being renewed to convince the domain of data practitioners to move to new solutions. Adoption and discussion within the RDA community can serve as a bridge to this larger adoption.

As indicated above we restricted ourselves in this phase of the work to define only those 10 core terms which have shown to find rough consensus. Many other terms that are being used and need to be tackled. A start on these can be found in the Appendix. One such example is the term "Research Data Object" that is being used in one model described in the model document as well as in the use case document. From the discussions it is obvious that this term is important in an RDA context but could not yet be defined adequately through to reach an adequate consensus. More discussions in the coming phases of term definition are required.

1.5 Method

For the purposes of this document and the RDA WGs we have engaged most closely we have chosen to define and relate a limited number of key terms. For immediate adoption we deem these relevant and useful to expose some core of data organization principles. This reflects some degree of synthesis based on an interpretation of the models and useful relations to ideas under development by other RDA WGs completing work at P4. The resulting snapshot concept-terms are described in a very simple language to encourage discussions, although we also provide some alternative ideas reflecting discussions as terms were developed. Such an approach seems to be useful (if not wise) after 18 months of discussing terms in all their different facets where most of us were tempted to leave the bounded world of our models and use cases. With this set of basic definitions we hope to encompass an essential and useful part of the RDA culture. It is important to note that this snapshot deals with registered digital objects and tries to cover the analyzed models. We recognize at this stage of the DFT WG effort that we need to:

- go out to communities for discussion and hopefully get adoption or agreement
- start elaborating on the definitions and bring them into a formal framework.

This however is beyond the current life of the DFT WG effort, although a start on a model synthesis has been developed as part of the Analysis and Synthesis documents. We expect that follow on work can be embedded in the RDA Data Fabric (DF) Interest Group, and/or possibly a follow on as part of a new DFT IG.

2. DFT Core Terms

2.1 Definitions

In this chapter we leverage some of the model analysis and synthesis and list all definitions in a compact form without further elaborations, since this is done via alternative definitions in chapter 2.2 and is also reflected to some degree by the definitions and discussions captured in the DFT term tool. We recognize that certain important nuances may be missing as a result of compaction such as the various aspects of metadata or the concept of representation information and information content under study by other RDA workgroups and the large existing body of work. More elaborated discussion of these concepts and others is also available in the DFT term tool.

1. **A digital object (DO) is represented by a bitstream, is referenced and identified by a persistent identifier and has properties being characterized by metadata.**

2. **A persistent identifier is a long-lasting ID represented by a string that uniquely points to a DO and that is intended to be persistently resolvable to access meaningful, current state information about the identified DO.**

3. **A PID record contains a set of attributes stored with a PID describing DO properties.**

4. **A PID resolution system is a globally available system that has the capability to resolve a PID into useful state information describing the properties of a DO.**

5. **Metadata contains descriptive, contextual and provenance assertions about the properties of a DO.**

6. **A digital aggregation is a bundle of digital entities.**

7. **A digital collection is an aggregation which contains DOs and DEs. The collection is identified by its PID and described by its metadata.**

8. **A digital entity is anything that can be represented by a bitstream.**

9. **A digital repository is an infrastructure component that is able to store, manage and curate DOs and return their bitstreams when a request is being issued.**

10. **A bitstream[9] is a sequence of bits that encodes a specific informational content, either stored on some media or being transferred under control of protocols.**

11. **State information is "metadata" information that describes those current properties of the DO that are relevant for proper management and access.**

12. **A property of a digital object specifies one of its characteristics.**

[9] Bit-Sequence and Bit-stream are seen as synonyms in the context of this work.

13. **A digital metadata repository is a type of digital repository that is able to store, manage and curate metadata.**

14. **A checksum is a type of metadata and an important property of a digital object to allow verifying identity and integrity.**

2.2 Elaborations

In this chapter we will list all definitions in their context of the major variants that have been mentioned and elaborate on them as part of DFT activity.

2.2.1 Digital Object (DO)

A. Definition

A digital object (DO) is represented by a bitstream, is referenced and identified[10] by a persistent identifier and has properties being characterized by metadata.

Note: As indicated we only talk about registered DOs in the context of this document.
Note: Properties included in metadata include discovery, contextual, schema, rights, curation and provenance information.
Note: A DO is said to be dynamic when the information content represented in a DO is changing for some period of time or even for indefinite duration.

B. Elaboration

There are many alternative views and definitions out there, we just want to mention 4 of them:

Variant 1

Digital objects (or digital materials) refer to any item that is available digitally. (Wikipedia)

Variant 2

A digital object is composed of structured sequence of bits/bytes. As an object it is named. The bit sequence realizing the object can be identified & accessed by a unique and persistent identifier or by use of referencing attributes describing its properties. (in DFT Term Tool and from the Practical Policy WG).

Variant 3

Digital Object is also called a Digital Entity defined as "machine-independent data structure consisting of one or more elements in digital form that can be parsed by different information systems; the structure helps to enable interoperability among diverse information systems in the Internet." (in DFT Term Tool)

Variant 4

The Fedora Commons architecture defines a generic digital object model that can be used to persist and deliver the essential characteristics for many kinds of digital content including documents, images, electronic books, multi-media learning objects, datasets, metadata and many others. This digital object model is a fundamental building block of the Content Model Architecture and all other Fedora-provided functionality. A Fedora object contains a persistent identifier. (Fedora Commons)

Variant 5

Digital objects are marked by a limited set of variable yet generic attributes such as editability, interactivity, openness and distributedness. As digital objects diffuse throughout the institutional fabric, these attributes and the information–based operations and procedures out of which they are sustained install themselves at the heart of social practice. (Kallinikos et.al.).

Variant 6

[10] Some repositories include passport like information with the PID which goes beyond pure referencing.

Digital objects consist of multiple elements, each of which consists of a type-value pair. Each of the types is represented by identifier and can thereby be interrogated individually. Identifying the data structure itself, instead of a specific file or folder that may contain it, or perhaps the machine on which it was first made available, enables persistent information access that is decoupled from most aspects of the underlying technology. (Robert E. Kahn: http://hdl.handle.net/4263537/5044)

Variant 7

A Digital Object is an entity consisting of a sequence of bits, or a set of sequences of bits, having an associated unique and persistent identifier. A DO may be static or dynamic, or some combination thereof. An entity is then defined as: An entity is anything that has a separate and distinct existence that can be uniquely identified.(Kahn et.al.)

It is important to note that not all communities insist on a PID and registration as definitional of a DO. But (a) in this document we are only making statements on the sphere of registered data and (b) many do and the value of that has been a theme in some of this work and that of other RDA WGs[11]. Fedora Commons, as cited in variant 4, offers an implementation of a specific DO model which is central to its architecture and allows users to bundle a number of content streams, to give it an identifier and to associate metadata descriptions with the bundle and its components which are themselves typed streams. So it fits with the definition we have chosen.

The first variant is a highly condensed view of DOs and may lack enough detail to support automating some aspects of DO management. The second variant specifies some additional metadata for a DO and also takes a more flexible approach to IDs recognizing the use of local IDs. The third variant is a process view in part, since it focuses on DOs' construction principle and its process characteristics, and does not tell us what about the structure is needed to enable interoperability. More information may be needed to help automate interoperability. The construction principle is reflected in the definition in an abstract way and the attribute descriptions within ID or metadata records will enable processing. The second variant makes use of the words "is being composed" instead of "is a". Since we find "is a" more simple and direct we opt for these words. Having a name, as is suggested in variant 2, is one of its properties that can be found in the ID and/or metadata records; therefore it does not to be specified in the definition. Variant 2 does not require an ID but also allows using "referencing attributes" as identification basis. Here, however, we would clearly like to speak of externally registered persistent and unique identifiers, since this will be the only way to register DO's as an explicit step as it is necessarily required in the domain of registered digital data. Variant 5 adds abstract and useful requirements which are essential for accessibility, but is neutral as to the role of an ID. It does not tell us how to document a DO to do this. Variant 6 refers to the internal structure of a DO and the importance of types that describe the elements of the DO independent of its creation contexts. In so far it comes close to the Fedora object model in Variant 3. Variant 7 includes the possibility of changing content, introduces the term "entity" and emphasizes the importance of being uniquely identified.

C. Conclusion

We can conclude that the above definition is in agreement with the 7 variants except for the explicit registration which will be essential in our growing domain of DOs.

2.2.2 Persistent Identifier (PID)

A. Definition

[11] This can be seen in analogy of the Internet. There are many nodes out there that do not have an IP address, but they cannot participate in the Internet exchange.

A persistent identifier is a long-lasting ID represented by a string that uniquely identifies a DO and that is intended to be persistently resolved to meaningful state information about the identified DO[12].
Note: We use the term Persistent Resolvable Identifier as a synonym.

B. Elaboration

There are a few definitions out there of which we want to mention 4:
Variant 1
A Persistent Identifier (PID) is a type of identifier, generally agreed is long lasting (non-corrupting) reference to informational content that distinguishes a digital or data object (...) from other Data Objects.
Variant 2
A Persistent Identifier (PID) is a string (symbol) that identifies a persistent digital object and that can be resolved to meaningful state information about the identified digital object.
Variant 3
A Persistent Resolvable Identifier (PRI) is a long-lasting string that uniquely identifies a DO and that can be persistently resolved to meaningful state information about the identified DO (such as checksum, access paths, references to additional information, etc.).
Variant 4
A Unique Persistent Identifier is a unique identifier that can be resolved to state information about the DO; and such identifiers may be separately branded. There may also be one or more identifiers associated with the information incorporated in a DO apart from the identifier of the DO itself. (Kahn et.al.)

Variant 1 is included in the definition above, but adds information to it which makes the definition slightly more complex and adds information that could be integrated in explanations. Variant 2 adds the aspect that a PID needs to be resolvable to useful information about a DO whether directly or indirectly which is essential for identity checking etc. Variant 3 includes this aspect of being "resolvable" in its name and thus makes its nature more obvious. The first part of variant 4 is included in the definition. The second part adds complexity which at this part we would not like to include.
It should be noted as can be seen from the models that some repositories are using URIs as persistent IDs to refer to a DO, but then use metadata to include crucial information for example to be able to check identity, i.e. state information is not directly associated with the identifier and some mechanism needs to be implemented to bind ID and state information. Thus, given that URIs are persistent they are PIDs.

C. Conclusion

We can conclude that the above definition includes the relevant core aspects. The acronym "PRI" describes better what we want to express, but the acronym "PID" is in broad use already. Both terms should be seen as synonyms.

2.2.3 PID Record

A. Definition

A PID record contains a set of attributes stored with a PID describing DO properties.

B. Elaboration

[12] This can be information such as checksum, access paths, references to additional information, etc. Some repositories call this administrative or system metadata, by NISO and others but some think this has not been well defined yet. In cases where the digital objects do not exist anymore, due to finite lifetime for example, the PID is expected to exist further and can still be resolved into useful information.

In use cases that are applying PIDs it is often mentioned that the PID record should at least contain path information to access the different instantiations of bitstreams and checksums of different types to allow proofing identity and integrity.

C. Conclusion

This definition covers the discussions in DFT and it is not specific in requesting the inclusion of specific properties. This is being discussed in the PIT WG.

2.2.4 PID Resolver (aka Resolution System)

A. Definition

A PID resolution system is a globally available infrastructure system that has the capability to resolve a PID into useful, current state information describing the properties of a DO[13].

B. Elaboration

Variant 1

A globally available system that has the capability to resolve a PID into useful information describing the properties of a DO.

Variant 2

As indicated above some are using URIs where the DNS system is used to make the path information actionable and others are using just the ID without a resolution mechanism associated with it.

Variant 3

An Identifier Resolution System is a globally or locally accessible system that has the capability of resolving a unique identifier to state information describing certain key properties of the DO or enabling access and use of the DO itself. (Kahn et.al.)

C. Conclusion

We can conclude that variant 1 definition is basically identical with the definition above, but does not express the essential aspect so well. Variant 2 is making use of the standard Internet resolution system which fits with the definition. It is worth noting that these definitions do not describe what the properties of a DO are. Specifying these as being started by the PIT working group is important to make DOs as useful as needed.

2.2.5 Metadata

A. Definition

Metadata contains descriptive, contextual and provenance assertions about the properties of a DO.

Note: Such metadata will make the DO discoverable, accessible and usable/interpretable.

Note: To make metadata referable it needs to be associated with a PID and thus is a DO.

Note: Metadata minimally needs to contain the PID.

B. Elaboration

Variant 1

Metadata is data that plays the role (is used for) data/resource discovery, description/ documentation and contextualization.

Variant 2

Metadata is a type of data object that contains attributes describing properties of an associated digital object.

Variant 3

Metadata ”represent the set of instructions or documentation that describe the content, context, quality, structure, and accessibility of a data set." Michener (2006)

[13] There are a couple of comparisons such as http://www.clarin.eu/content/comparison-pid-systems

Variant 4
Metadata can be representation information which is information that maps a data object into more meaningful concepts.
Variant 5
Metadata is a description, which could itself be managed as a separate DO, which consists of assertions about a DO in order to make the DO findable, accessible, interpretable or otherwise usable. The description, at a minimum, must contain at least one searchable term other than its unique persistent identifier. (Kahn et.al.)

Metadata in the form of key-value pair descriptions of properties of digital objects is an old concept first being defined in the library community and then taken up in many scientific communities. It has been used to describe all kinds of properties by different sub-communities. DFT necessarily needs to stay with an abstract definition that clarifies the important role of metadata in basic data organizations, but for now leave clarifying details to the various MD IG & WGs. Variant 1 definition is very unspecific in terms of data organizations but gives examples of widely agreed roles. Variant 2 is basically identical with the chosen definition. Variant 3 and Variant 4 are both addressing the essential task or role of metadata descriptions in different words: describe properties of a DO. Again this may be too general to be useful so additional work is needed in the metadata groups of RDA. Variant 4 makes use of a more abstract type of definition. The Variant 5 definition above expresses that a metadata description is a type of data object which is essential for data organization considerations. It also expresses essential requirements of describing object properties in a not too abstract way but without details of structures to satisfy these requirements. It adds the sentence that for the registered domain of data it is important to have at least the PID included somehow in the metadata which is important for proper data organization.

C. Conclusion
We can conclude that the definition includes useful requirements for data organization and its function but does not specify structural and semantic details. The one structural inclusion noted is that for data management a minimal MD description should contain the PID of the described DO. Recommendations on how a DO is described are not addressed in these definitions. A widely accepted requirement is that in particular metadata syntax and semantics must be registered explicitly to foster integration and interoperability.

2.2.6 Aggregation
A. Definition
A digital aggregation is a bundle of digital entities.
Note: The term "aggregation" as a base concept does not add substantially to our understanding of the intuitive idea of collections as resulting from some aggregation process and thus is not used as a separately defined concept.

B. Elaboration
Variant 1
An aggregation is in general the bringing together of elements.

Variant 1 definition introduces the undefined term “element” and does note express the structural composition as a basis of an aggregation as crisp as the definition above. Element may be a more specific term about parts than the idea of “bundle” which about a whole used in the first definition. Understanding whole from parts is usually the way we proceed such as building a collection from part. Types of aggregations differ by the nature of the processes by which elements are brought together and the reason understood for aggregating or contained as a whole unit.
Important to note is that the term "aggregation" includes the possibility of bringing together digital object as defined above as well as any other digital entity which has not a PID assigned and/or a

metadata description. This makes it possible to give collections state of a DO although their components are not DOs.

C. Conclusion

Some idea of parts making up an aggregation may be needed and relations between parts may need to be covered as part of metadata descriptions. The definition in terms of a structured bundle is agnostic as to the nature of the structure. Neither definition of aggregation makes direct statements about being registered and described by metadata; it just describes its construction principle in an abstract way.

2.2.7 Digital Collection

A. Definition

A digital collection is an aggregation which contains DOs and DEs. The collection is identified by a PID and described by metadata.

Note: A digital collection is a (complex) DO.

Note: A digital collection is a type of an aggregation in so far as there are other types of aggregations.

B. Elaboration

Variant 1

A collection is a form of aggregation of elements that has an identity of its own separate from the identity of the elements.

Variant 2

Collection is defined as a "group of objects gathered together for some intellectual, artistic curatorial purpose.

Variant 3

A digital collection is a type of aggregation formed by a collection process on existing data and data sets where the collected data is in digital form.

Variant 4

Collection is a type of aggregation obeying part-role relations and is a digital object since it has a PID to be referable and metadata describing its properties.

Variant 5

A Digital Collection is an organized aggregation or other grouping of distinct DOs that are related by some criteria and where the collection is described by metadata. A Digital Collection may also be identified by a unique persistent identifier, in which case the collection may be construed as a DO. (Kahn et.al)

Variant 2 focusses on the purpose of aggregating data objects which is an important aspect but not relevant from a structural point of view. But it may be useful here as a placeholder since no other structural view is provided. Variant 4 introduces a part relation aspect but beyond that does not add essential information as to the nature of the relations. Variant 3 describes the process of creating a collection which does not define what it is. Variant 1 specifies what a collection is. Variant 5 makes a distinction between aggregations and other groupings which are included in the above definition, since aggregation is not constrained. Since we only speak about the domain of registered DOs, the second half of the statement in variant 2 does not need to be included in the definition. The definition above combines the description of what a collection is and indicates that a collection itself is a digital object which is relevant in terms of data organization aspects, but it remains within the limited term space.

C. Conclusion

We can conclude that none of the definitions above includes structural characteristics/organization beyond the idea of an associated PID and that it has some metadata to describe its content, etc. This

allows repositories to implement the collection mechanism in any suitable way. In this sense the definitions do not handle the details of aggregations as in the OAI-ORE which has a resource map construct to detail the components of an aggregation or digital collection or as in the CLARIN case where just another schema based metadata description specifies the content. Important in all solutions is that schema, element semantics and relational semantics are made explicit to enable machine based interpretation.

2.2.8 Digital Entity

A. Definition

A digital entity is anything that can be represented by a bitstream.

B. Elaboration

Variant 1

Digital Entity definition from X.1255 ITU standard "machine-independent data structure consisting of one or more elements in digital form that can be parsed by different information systems; the structure helps to enable interoperability among diverse information systems in the Internet.

C. Conclusion

The X.1255 ITU definition is more elaborate than the definition found here which is kept on purpose simple.

2.2.9 Repository

A. Definition

A digital repository is an infrastructure component that is able to store, manage and curate DOs and return their bitstreams when a request is being issued.

B. Elaboration

Variant 1

Repository is a searchable and queryable interfacing entity that is able to store, manage, maintain and curate data/digital objects

Variant 2

A repository is a location that is able to store, manage, maintain and curate a digital object and return its bit-streams if a persistent identifier is being issued.

Variant 3

The OAI repository is a type of repository with a network accessible server that can process the 6 OAI-PMH requests in the manner described in the OAI implementation guide.

Variant 4

A Repository is a digital collection organized to provide specific DO information management services to users, such as storage, identification and performance of operations on a DO including returning a DO in response to a request. A repository may itself be considered a complex DO that incorporates one or more DOs

We need to accept that repositories (of digital objects) can specialize themselves by focusing for example on metadata, on data or even on software components for example, therefore we need a definition that is not specialized as it is indicated for example in variants 1 and 3. An alternative approach is to have separate concepts by type of repository which can be done in a second step.
While it seems desirable and in practice, a repository does not have to provide a searchable interface since it could submit all its metadata for example to a metadata harvester which will then offer search capabilities, however it must offer content in form of bitstreams once this is requested. Also we cannot expect that the access interface will expect a PID, since in general a PID resolver generates path information that allows accessing the object.

C. Conclusion

We can conclude that it is important to mention a repositories basic function - namely taking care of digital objects stored and delivering its content on request. Policy to specify some of the data management functions of a repository may be provided by the PP WG. We add that in an abstract sense a repository could be seen as a DO[14].

2.2.10 Bitstream

A. Definition

A bitstream is a sequence of bits that encodes a specific informational content, either stored on some media or being transferred under control of protocols.

B. Elaboration

Variant 1

Bitstream denotes an unstructured sequence of bits that is identified as unit. (A digital object may be represented as bitstreams of finite length that encodes informational content).

Variant 2

A bitstream is the transmission of bits as a simple, unstructured sequence of bits or the sequence and content of such a transmission.

Variant 3

A bitstream is a sequence of bits that is being stored on a storage medium or that is being transferred via some kind of network. A bit stream is encoding some form of data, information or knowledge.

Variant 4

A bit-sequence is sequence of bits (1s and 0s) that may or may not have any meaning. (Kahn et.al.)

Variant 1 makes the point that some decision is made as to the bitstream as a unit. Variant 3 expresses the essential aspect in terms of data characterization: it represents encoding data, information or knowledge (if we agree that algorithms for example are expressions of knowledge). In the context of data organization we also want to express that bitstreams are the medium carrying potential content that a digital object has and thus that it has a finite length in relation to digital objects. This makes them different from continuous bitstreams that are generated by sensors for example. Once becoming a part of the registered domain of data these endless streams need to be subdivided into structural and manageable units as noted in Variant 1. Dynamic data, i.e. sensor streams, where its fragments do not appear in a synchronous way, form special challenges which are not being discussed here. Variant 2 does not really help in our discussion since it emerges from computer networking, however, in data organizations we need to understand that bitstreams that encode the requested information are not only exchanged/transferred via networks, but also via any kind of suitable protocol. Variant 4 stresses the point that bitstreams at first instance are just a sequence of bits and that they upfront do not to have a meaning. However, in our domain of registered data we expect metadata describing the content which in theory could be "meaningless big-sequence".

C. Conclusion

We can conclude that with the addition of the unit idea the definition above includes the essentials of being a sequence of bits encoding requested content, that it can be stored and preserved and exchanged via protocols. It should also indicate for our term landscape that DOs have encoded content and that this is of finite length: a DO has bitstreams of finite length.

[14] A repository also needs to have much management operations etc. in place which extends it beyond a mere DO, but actually all DOs stored in a repository can be seen as one big collection.

2.2.11 State Information

A. Definition

State information is "metadata" information that describes those current properties of the DO that are relevant for proper management and access.

B. Elaboration

Variant 1

State Information is relevant and current actionable information about a DO such as its current location(s), public key(s) and other validation information. (Kahn et.al.)

The term state information is widely used by some data practitioners as a neutral term that does not include the many different metadata terms that have been used by sub-communities. Terms such as system metadata and administrative metadata are in use for example. In the above definition we simply speak about metadata that is relevant for management and access, thus metadata that does not describe properties related to its content, although there is no clear boundary. For curation processes one would need to know about the type of the object.

C. Conclusion

We employ this concept to note that metadata must include some current information about a DO's state such as location. This type of metadata is typed here since a valid set of such information seems necessary to make the use of PIDs work. As needed other types of metadata may be specified for different data management functions, but these are not yet discussed in this snapshot.

2.2.12Property

A. Definition

A property of a digital object specifies one of its characteristics.

B. Elaboration

Variant 1

Attribute is a characteristic of data that sets it apart from other data, such as location, length or type. The term attribute is sometimes used synonymously with "data element" or "property." (reference: http://www.krollontrack.com/resource-library/glossary/legal/#d)

Variant2

See the definition above where a distinction is made in the following sense: an attribute is a formal constituent of a structure which can contain the specification of a property of a digital object.

Variant 3

A Property is the smallest, atomic part of metadata written and read by the PIT API. It is defined in the Type registry. It consists of a number of elements.

All variants of object properties capture the idea that there are distinguishing characteristics of the digital object. Variant 2 and thus the definition states this point in a very simple way. Variant 3 was introduced by the PIT group for their specific needs and is perhaps more detailed than what DFT needs to express for general use. It is, however, included in the elaboration since it refers to properties often stored in the PID record in various use cases for example as atomic characteristics. It is also useful to note that a PID record may be thought of as a Property Record, which is listed in Appendix B.

C. Conclusion

The definition is generic enough to cover PID record and metadata description purposes and can be aligned with the more specific PIT definition.

2.2.13 Metadata Repository

A. Definition

A digital metadata repository is a type of digital repository that is able to store, manage and curate metadata.
Note: A metadata repository is a type of digital repository, i.e. it is associated with a PID.

B. Elaboration

The idea of a metadata repository is included here for completeness and as a placeholder for subsequent expansion by the various metadata groups. The work of the various metadata groups, for example, may expand the placeholder idea of metadata repository with further distinctions between catalogs, directories and repositories based on such things as functions and services. Initial discussion indicates that there is enough of a distinction to note here that some separate consideration of repository services for metadata may be important. Separate policies for these, for example, have been noted in the work of the PP WG for such things as metadata extraction.

C. Conclusion

Due to its importance also for many other RDA WGs and IGS we added this definition. The metadata WGs/IGs need to work on details.

2.2.14 Checksum

A. Definition

A checksum is a type of metadata and an important property of a digital object to allow verifying identity and integrity.

B. Elaboration

The term checksum is a widely used term. A checksum is a randomly generated piece of data calculated by some algorithm that is used to verify the fixity or stability of a digital object. It is most commonly used to detect whether some representation of digital object has changed over time.

C. Conclusion

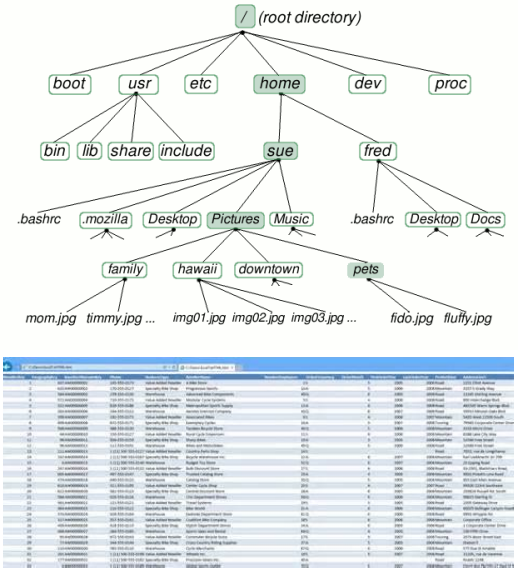
The definition is widely agreed and does not need argumentation.

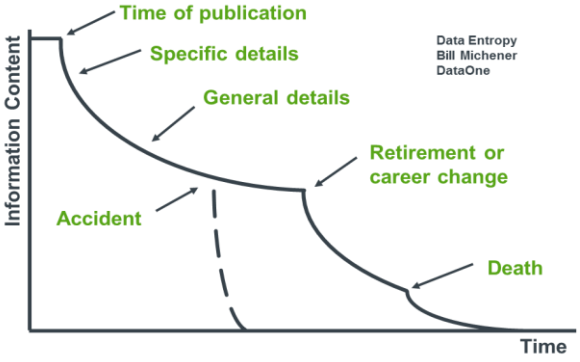
Appendix A: Concrete Examples of Data Practice

Our conceptualization needs to be based on concrete models as described and analyzed in our model papers. The 22 models we cited are well thought-through models in terms of proper data organizations, however, there are many models that emerged through ad hoc discussions or which follow traditional practices. Data created according to these models cannot easily be included in our agreed-upon open domain of registered and sharable data objects. We will discuss here three representative examples.

File System Usage

File systems are the "natural store" for data for many years. The software that runs file systems is robust and generally reliable, and its features are well-described. File systems are available on all computer platforms, although many are now hidden behind powerful cloud interfaces[15] because of limitations with respect to performance and addressing capabilities. There is a long practice to store data of different sorts in file systems. The file and directory names were used to encode essential metadata (time, creator, session, etc.) and smart schemes of abbreviations have been invented. The directory structure was used to organize the data into meaningful collections. Recently, we have started to recognize the many limitations of file systems, especially when aggregating lots of data, when creating new data out of existing data using a variety of operations, when combining data to new collections to work on, and more. These limitations mean the scientific goal of reproducible data is jeopardized because researchers cannot oversee what they have done after a period of time. Michener uses the term "data entropy" to describe the situation in many labs. For example, when someone leaves a lab or research facility, it is often impossible to trace the work they have done and the data they have compiled. One reason for this problem is that file names cannot adequately document context and provenance and cannot document all relations among the different files that have been created.





This figure includes a graphic from B. Michener about "data entropy" indicating the decreasing knowledge about our stored data. It also indicates typical file system organizations and an indication of a typical spreadsheet solution to cover metadata.

In addition to file systems, researchers began using spreadsheets to document their work and the structure of their data, and some also use relational databases[16]. This is certainly a step forward, since a spreadsheet can include many properties about the stored data. Spreadsheets are widely

[15] Cloud systems are defined by a simple interface where a hash code indicates the "object" being addressed, but all "logical information" describing properties of the "object" are managed by an application, i.e. only the applications on top of cloud systems address the issues being discussed here.

[16] This should not be mixed up with large repository systems that make use of powerful relational databases to store so-called system metadata that is used to manage huge multi-layer file systems for example.

used because they are easy to handle, include filtering and sorting operations and offer many other features that make them useful. However, spreadsheets have two major drawbacks: (1) In general, there is no agreement on what to encode and how to encode it, in other words researchers or projects invent their own encoding schemes; and (2) there is no "genuine binding" with the data stored, which means in practice that spreadsheets get lost. This is a problem considered to some degree as a Use Case Scenario for Research Object in the DFT Use Case document. The use of spreadsheets improves the situation somewhat but experience shows that information about the stored data will still be lost. In fact, the loss of information is often worse because people relying on spreadsheets do not bother to encode information in the file names. These are problems to reflect the fact that spreadsheets stored outside of repositories do not make metadata documentation easy.
The situation becomes even more complicated when people exchange data. Spreadsheets are rarely exchanged and when they are, they are often impossible to understand by other people or machines.

Database Usage

Often people are using database management systems (relational, XML, etc.) to store data. In general a "logical structure" specifies the content of the included structures, meaning the structure is documented and the information is being constrained. Usually these database management systems use data encapsulation to store data in the most optimal way, but that can result in characteristics that make it harder to address specific objects. Specific issues include the following:

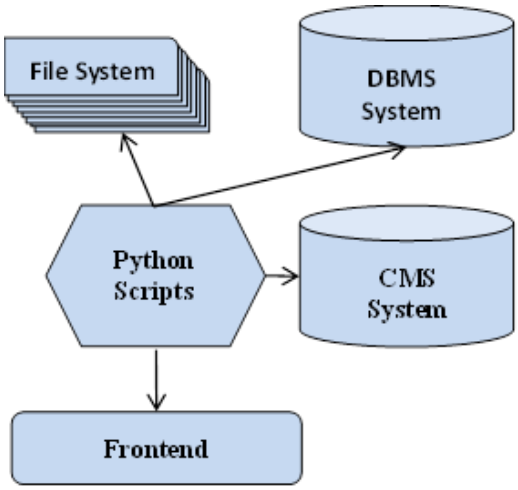
- To access bitstreams that encode specific content, the researcher needs to have an application on top of the database. Others than file system software these applications change often, are maintained by different people, etc.
- Often, the content that a researcher wants to access as a unit of information is not contained in a single cell (in the case of a relational DBMS). Instead, storage of that unit of information is spread among a number of cells and tables. This happens when a join operation needs to be carried out and the "object being addressed" is created dynamically.
- DBMS are designed to store a wide variety of contributions from different persons or experiments and they are optimized to allow for dynamic changes. To assure that citable content does not change over the years and reproducible science is possible, applications must be developed that ensure proper separation and proper versioning. All this can be done in a DBMS, but the software must be written and maintained by the researchers. Separation can usually be done by the DBMS; however, preventing changes to existing content requires software development. Often, researchers do not have the time or expertise to carefully design and maintain these kinds of software systems.

Basic file system properties do not change so researchers can be assured that their addressing mechanisms and simple operations will prevent changes. In a DBMS, a layer of complexity is added to the software stack that requires careful design and maintenance. This means that defining a digital object encapsulated in a DBMS also requires identifying the version of the software code in order to extract the exact intended content.

User Defined Repositories

Researchers who collect data of some particular type often create their own software frameworks to make the data accessible, and their work has the best intentions. At every scientific meeting, there are examples of these kinds of solutions, which are tailored to serve the immediate purposes of the scientist, most often by creating software components that need to be maintained. Here, we provide one such example, further elaborated on in the figure below. We also discuss the pros and cons of this solution.

Typically larger chunks of information are stored in file systems, and smaller amounts of data and some structured information are stored in a relational DBMS such as mySQL. Often, Python scripts are used that include procedural code as well as code that specifies many kinds of relations between the different chunks of information. Together with Python scripts often some Content Management System is used to also store some metadata and relational information. These mixed systems as indicated in the figure have been designed to easily add content and post it on the web to make it accessible. The major function is taken care of very well, but the management and federation of such data is not as easy and requires major efforts, in particular if these systems have a considerable size and complexity.



This figure indicates a typical data solution found in many institutes and projects.

A simple operation, such as replicating the data and information contained in such mixed systems, already requires physical programming efforts, such as replicating the files, and replicating the DBMS and the CMS. Even then, relational information and metadata encoded in the scripts will be lost. Federating such data, including the logical information that is essential for re-use and cross-disciplinary discovery and use at later dates, requires enormous efforts, as the DataONE and EUDAT projects learned when integrating data from different repositories created by experts in different fields. Data integration challenges exist not only at the level of semantic interoperability—where term definitions are often lacking—but already at the layer of data organization. The ways in which research labs, projects and scientific disciplines store data, the kinds of metadata they employ, and in particular the relationships between the data and metadata differ substantially. In EUDAT, for every center that wanted to join the data federation it was required to develop software to integrate the data. That software requires maintenance, which means federating data is currently non-scalable and too costly for all but the largest collaborative projects. The result is that in practice, most often only physical-level data solutions are carried out and all logical layer information is lost.

Appendix B: Additional Terms

From the snapshot core view focusing on the work of early RDA WGs fon DO PIDs etc. these terms are not as essential for a conceptualization of a proper data organization. But some have been discussed at prior meeting (P3) and posted to the term tool where discussions were held. Therefore we will mention them here indicating something about their nature (e.g. what type of thing they are), purposes (e.g. allows data management) and inter-relationships (e.g. an identifier is associated with essential metadata). In some cases there is nothing besides placeholder such as pointing to a WG that will/should define the concept. However, the short sentences should not be seen as definitions, but as rough indications of meaning. More discussions are required to come to a larger core view of term definitions and relations between them.

Active Collection	denotes a collection that is being generated at access time
Active Data	denotes data that is being generated at access time
Container	is a structure that allows bundling of DOs
Data Citation	to be defined by the citation groups
Data Lifecycle	describes the processes a DO will undergo from its creation
Data Object	is a type of DO containing processible data/information/knowledge
Data Organization	denotes how DOs of different types (metadata, data, collections), their relations among them, the access information and other types of metadata are structured and managed
Data Publishing	<to be defined by the publishing groups> [17]
Data Set	is a more abstract notion of a managed aggregation of DOs
Data Stream	is a generalization of the term bit-stream
Dynamic Data	specific type of data being transformed to become DOs
External Property	those properties that allow management and access of a DO (state information)Identity is a string associated with a DO to refer to it
Information Object	is a bundle containing the DO and its metadata
Integrity	is a statement about the accuracy, validity and consistency of a DO
Internal Property	those properties that allow to interpret the content of a DO
Landing Page	web-site with more contextual information about a DO
Object Property	is an element characterizing an aspect of a DO stored in MD and/or PID records
Original Repository	is that repository where a DO was deposited first
PID Attribute	is the formal element of a the PID record
PID Information Type	is the type of a PID attribute
PID Record	is the set of attributes stored with a PID describing DO properties
Presentation Version	is a metadata attribute describing the functional purpose of a DOs' version
Property Record	is a generic term covering, for example, PID records
Real-time Data	specific type of data being transformed to become DOs
Research Data Object	A research Data Object or Research Object (RO) is, or provides, a container for a principled aggregation of resources, produced and consumed by common services and shareable within and across organisational boundaries.
Service Object	is a type of DO containing executable code
Transaction record	a data structure that contains the history of access to a DO

[17] In the term tool this is defined as: The process whereby data are subjected to an assessment process to determine whether they should be acquired by a repository; followed by a rigorous acquisition and ingest process that results in products being publicly made available and supported for the long-term by that repository.