**Data Type Record – Elements for Characterizing Data**

**1. Note to ISO DTR Study Group Members**

* This document specifies elements found to be useful and important for characterizing data managed in projects by CNRI and other Research Data Alliance (RDA) members. Multiple instances of Data Type Registries are currently in use or being evaluated by the RDA members. An ISO standard in this area would be encouraging and fruitful.
* Please see which of these elements are currently being supported or described in existing ISO standards, and which of these are new. And also deliberate on next steps for the DTR study group activity. Our next virtual meeting is in March 2017. Please be prepared to discuss your views in that meeting, but I encourage everyone to discuss this topic via email prior to that meeting.
* I have simultaneously shared this document with the RDA group. I will send updated versions of this document to reflect any additions or changes I receive from them or from early feedback from ISO Study Group Members.
* I deliberately did not define a concrete schema for the data type record in order to keep the discussions focused on the elements themselves versus how they are being represented.
* Also, I strongly suggest avoiding debating about terminology (for now, anyway).

**2. Example and Context**

This document introduces the concepts of Simple Data and Complex Data prior to describing what elements may be used for effectively characterizing such data. Note that these concepts of Simple Data and Complex Data are not terms of art, but are merely used here as tools for laying out the groundwork for what elements could characterize data in a useful fashion.

Simple Data and Complex Data are interrelated, as shown in the example in Figure 1. Simple Data is a single value captured in any dataset, e.g., a temperature reading, or a pressure reading, or a timestamp.

Complex Data can be thought of as a collection of Simple Data items in structured form. For most tabular Complex Data, all data items in a single column will be of the same Simple Data type. Though the Complex Data example presented here is of tabular form (examples of which are CSV and NetCDF), Complex Data may instead be hierarchical (e.g., HDF5), or assembled from a set of files (e.g., XMLs or JSONs).



Figure 1: Example of Complex Data and Simple Data Descriptions

**3. Simple Data Type**

A single piece of data, aka datum, e.g., a number or a string, can be characterized using one or more of the following elements:

1. **Primitive**: Boolean, Integer, Float, String, Date, Time, DateTime, and Spatial. (Alternatively, Date, Time, DateTime, and Spatial can be included with Concepts element as stated below).
2. **Scope**: The set of constraints to define allowable values:
	1. Enumeration: An enumeration of allowable values.
	2. Regular Expression: A regular expression to codify allowable values.
	3. Range: A range of values applicable for integer, float, date, time, datetime, and spatial types.
3. **Language**: The natural language in use for specifying the value (if applicable), e.g., English, Mandarin, Japanese, or French.
4. **Presentation**: Any presentation and formatting patterns, e.g., date values are expressed as YYYY/MM/DD.
5. **Concept**: Concepts could be quantifiable or otherwise.
	1. Quantifiable concepts define measurable entities such as distance, pressure, and temperature. They also include ratios and factors such as distance/time or distance-squared.
	2. Non-quantifiable concepts include Tree, Bridge, Person, House, (Date?, Time?, Spatial?), etc.
6. **Measurement Unit**: For values associated with quantifiable concepts, measurement units may also be associated, e.g., Fahrenheit and Meter-Squared.
7. Other useful elements:
	1. **Description**: Describe this data type record in a human readable fashion.
	2. **External References**: List of existing standards and recommendations that describe the datum of interest either in part or in full.
	3. **Provenance**: A structure that captures who created this data type record, when, and possibly why.

**4. Complex Data Type**

A complex piece of data, such as structures that have data in nested form, tabular form (CSV, NetCDF) or hierarchical and assembled forms (HDF5, XML, JSON), can be characterized using one or more of the following elements:

1. **Properties**: An (ordered) enumeration of identifiers that provides a structural breakdown of the Complex Data.

In the case of tabular data, each row is a collection of Simple Data values. As such, this (Properties) element captures the ordered list of Simple Data Types – one for each column. A separate flag to indicate that tabular data consists of multiple rows sharing the same Complex Data Type is specified as an attribute. Additional attributes may be included as necessary.

1. **Dependencies**: A construct to express dependencies across properties.
	1. **Functional Dependency**: In cases where a value is a derivative of other values or at least the presence of one value is dependent on the presence of other values, e.g., volume is a derivative of length, width, and height, such a dependency can be expressed here. NetCDF based datasets express these dependencies.
	2. **Cardinality Dependency**: Example -- Property 1 (date) exists only if Property 4 (pressure) exists.
2. **External References**: References to pre-existing standards or ontologies that describe the data, e.g., RDF/OWL.
3. **Services/Software**: One or more references, with necessary specifications, to existing Internet services and/or software packages that can process data pertaining to this data type. Such services or software could take as input datasets pertaining to one or more data types and produce visualizations and/or data of another type(s).
4. Other useful elements:
	1. **Description**: Describe this data type record in a human readable fashion.
	2. **Provenance**: A structure that captures who created this data type record, when, and possibly why.

**5. Resolvable Identifier**

Each Data Type record, whether Simple or Complex, will be associated with a resolvable identifier that resolves to the data type record (managed in a data type registry). The identifiers are also used for composing Complex Data Type records, specifically in the Properties element.

**6. Data Types and General Metadata**

Metadata is usually associated with datasets. ISO 11179-7, for example, in draft form, specifies metadata models that can be used for describing any given dataset “instance”. For example, it may be the case that sensors located in Reston, Virginia may have generated the dataset example shown in Figure 1, and that the specified hardware used for taking the readings is Davis Vantage Pro 2. Such information is usually recorded using ISO 11179-7 or equivalent models. That information, however, is deliberately kept out of the Data Type records.

The general idea behind Data Type records is to capture information about data that is applicable across “multiple” instances of datasets regardless of, in our example case, the location, hardware, date, and time. That is, the Data Type records characterize data, but in a way that is independent of the specifics of any given dataset instance. The reason for this is to encourage reusability of data types across environments, communities, and scientific disciplines.

**7. Standard Constructs for Data Type Reusability**

Complex Data Type could be defined as a composition of existing Simple Data Types. However, other constructs besides composability are found to be useful. An analog of constructs available in programming languages, such as sub types, extended types, and aggregated types are preferred instead of their reinvention.