Documenting Data Structures

Applying the DDI 4 Data Description to Various Types of Data Structures

Draft 2019-04-26

# Overview

The DDI 4 Data Description provides the basis for describing a broad range of data structures.

This document describes the general approach which DDI 4 is taking, and then goes through the details for a selected set of data structures. The goal for DDI is to describe various data structures, both legacy like rectangular data sets, multi-dimensional data, event data, also new ones like data streams or data lakes. The approach is independent from a specific domain.

The purpose of this document is the following:

* Documentation of things that the DDI 4 Data Description can handle.
* Communication of the DDI 4 Data Description to people involved in the DDI 4 development and to the broader DDI community.
* Analyzing the current DDI 4 model in order to detect possible gaps and to describe requirements for the refinement of the model in the MRT project around DDI 4 Core.

# DDI 4’s General Approach to Data Description

The DDI 4 Data Description is based at the core on the description of a single datum or data cell.

A Datum, like for example ‘73,3’ in figure 1 below, is basically a value that represents an instance of data that has been collected, derived, scraped, etc. A Datum has a meaning and populates a cell of a dataset, database table etc.[[1]](#footnote-1). The general idea in DDI4 is to be able to attach all necessary metadata to the single Datum so that it can be ‘followed’ across different data structures. This differy from some other approaches used in other DDI products (DDI Codebook, DDI Lifecycle) where some of this information was attached at a higher level (eg, the data set or record).



*Figure 1. Datum and it’s connotations*

# Data Description of Various Types of Data Structures

Data structures are a way to organize data in a structured way in order to be processed by software programs. The task as defined for the NADDI Sprint has been to explore how the current DDI4 model can be used to describe data from different data structures using the Datum based approach, as well as to identify gaps in the model that needs to be filled.

During the Sprint, examples for the following data structures and cases were documented:

1. Unit record data structure (wide format), based on examples from W3C recommendations[[2]](#footnote-2).
2. Tall data format (often used to express event data)
3. Unit record data transformed into a Tall data format
4. Cube (multi-dimensional or aggregate) data transformed into a Tall data format
5. Unit record data transformed into Cube data
6. Context-specific (“Viewpoint”) roles of variables and their usage in different formats
7. Example from data based on the RAIRD information model

Examples 1 – 7 will explained in detail below. Open issues related to the model that have been detected, as well as further extensions to the ongoing tasks will be described at the end of this document.

## Examples

### Unit record data structure (Wide format)

A Unit record data table, as shown in Figure 2, is a common way to organize data. This structure is also referred to as a rectangular data file.

###

*Figure 2. Unit record data table*

A cell in the Unit record table is an intersection between a column representing a variable and a row representing a measurement unit. Each cell of the table contains a Datum.



*Figure 3. Unit record data cell*

The objects of the Wide format Unit record data table are Unit records, Variables and Values.

 In the Wide format the rows correspond to each unit record, which is a set of values for one entity. The columns correspond to each variable measure or categorization. Cell entries are values.



*Figure 4. Wide format objects*

In figure 4 above ‘Marie’ and ‘Henry’ are identifiers for each of the records. ‘Sex’, ‘Longevity’ etc. are variables and ‘Female’ and ’73,7’ are example of Datum values.

### Tall data format

The same data can be expressed in a different format called Tall as shown in figure 5 below. This format is often used to express event data.

In the Tall format columns correspond to each kind of object in a Wide (unit record) description. Each row now contains a Unit Row Identifier, a Unit Variable, and a Unit Cell Value.

The rows correspond to each value of each (non-identifying) variable for each Wide record.



*Figure 5. Tall format – often used for event data.*

References to other DDI objects are needed: Identifier to each row in Wide format, Reference to each variable from Wide format and ability to handle every kind of value in one column (mixed datatype).

### Unit record data as transformed to a Tall format data

Figure 6 below shows the mapping between the Wide Unit record format and the Tall format. We see that all combinations of variables and values for each Unit record identifier are retained. Each value in the record for Marie now has its own row, with a second value – the Unit Variable – telling us what the value is (the column in the Wide table). The Unit Cell Value is the Datum.



*Figure 6. Transformations from Wide to Tall format.*

### Cube data (aggregate or multi-dimensional data) transformed to Tall format

Cube data are multi-dimensional data that can also be expressed in the Tall format as shown below.

###

### *Figure 7. Cube data transformed into a Tall data format*

### Unit record data transformed to a data Cube

Unit record data can be transformed to Cubes (aggregate/multi-dimensional data). Data from the individual units contribute to the aggregates of a Cube. We see that ‘Mary’, ‘Henry’ and the others contribute to the aggregate statistics of the Cube. The appropriate Unit record datum are averaged, producing the datum for the Cube cell. In the Cube below Marie contributes to two different cells due to overlapping time periods, while Henry only contributes to one cell.



### *Figure 8. Unit record data transformed to Cube data*

### Contextual (“Viewpoint”) roles of variables and their usage

Viewpoint roles are a very powerful feature that allows users to assign different roles to variables according to their context of use. Viewpoint roles are not inherent in variables but can be imposed on them. In DDI4 there are currently three roles:

* + **Identifier** - An identifier role that serves to differentiate one record from another. More than one variable may be used in combination to produce a compound identifier.
	+ **Measure** – Variables tagged with the measure role have the values of interest.
	+ **Attribute** – The attribute role serves to provide information about the measures of interest. Variables might, for example, describe the conditions of a measurement. This way attributes can be used to link metadata or paradata to the Measure of interest.

A variable may take on different roles in different contexts.

In the figure below **PersonID** is an identifier for a person, **Sex, Born, Died,** and **Longevity** are the measures of interest and **RefArea** might be considered an attribute of the measures.

These roles are not fixed. For another purpose **RefArea** and **Sex** might be the measures of interest. Roles are often slightly different when the same data is viewed using different formats (PersonID is the only identifier needed for the Unit Record format below – when expressed in a Tall format, it would be only one needed component of a compound identifier – more than one variable would take on the role of identifier. (See RAIRD example under point 7.).



### *Figure 9. Viewpoint roles of Variables*

### Example from DDI4 usage in the RAIRD information model

The example bellow shows what a possible dataset based on the [RAIRD information model](file:///C%3A%5CUsers%5Cuser%5CDownloads%5CThe%20RAIRD%20information%20model%20is%20an%20example%20of%20real%20life%20usage%20of%20the%20Variable%20Role%20feature).

RAIRD uses a mix of Tall and Wide layouts in that they add StartDate and EndDate as attributes that identifies a value. In figure 10 we recognize the crosswalk from the Wide Unit record data format to Tall, as explained in figure 6. StartDate and EndDate variables for each value are added additionally.

The keyValue table expresses the collection of variables in a possible RAIRD data set and how they are ordered. Key values link roles to each of them.

 

### *Figure 10 Example from the RAIRD information model*

Here, we see that both CaseID and VariableRef function as identifiers – taken together, they uniquely identify a record in the Tall format, and indeed as the identifier for a specific measure (the Value).

# Outcomes of the task: identification of gaps

When we apply the current DDI 4 model to these data descriptions, it becomes apparent that some gaps remain when it comes to the documentation of event and aggregate data structures.

* For the Value columns of the Tall format (see Figure 5), a generic “superset” datatype is needed in order to describe Datums from the different variables of the corresponding Unit record data file.
* For the VariableRef columns of the Tall format (also Figure 5), Value Mappings are needed.



*Figure 11 Tall layout*

* When Viewpoints are applied to Data Cubes, a **dimension** role needs to be added.
* Currently a variable can have more than one role in the same Viewpoint. Restrictions should be put to limit the number of roles a variable can play in the same Viewpoint to one.

Proposals for solutions to fill these gaps have already been formulated and are expected to be implemented in the model in the immediate term.

## Post-Sprint follow-up work related to the task

In addition to the implementation of the proposed changes to the model, a follow-up task to the work performed at the NADDI Sprint is to relate each of the examples to the UML components used to resolve them, laid out in a pedagogic way. This will provide documentation of the correspondences between the examples and the model, and the terminologies used in each of them. The planned task will also serve as an additional quality check of the model.

One idea is to use the GSIM presentation as a guide – they use high-level diagrams and other simplified presentations of the UML to make it accessible. We may wish to use a similar approach.

## The Variable cascade – a powerful feature of DDI 4

The above examples shows how a Datum should be able to flow between different data structures and how Viewpoint roles can be applied to variables. Another powerful feature of Data Description in DDI is the Variable Cascade that facilitates distinctions between Instance Variables (the variable as in the dataset), Represented Variables (the reusable components of a variable) and Conceptual Variables that expresses the conceptual basis of a variable. The powerpoint [DDI Variables and variable cascade introduction.pptx](https://ddi-alliance.atlassian.net/wiki/download/attachments/689242207/DDI%20Variables%20and%20variable%20cascade%20introduction.pptx?version=1&modificationDate=1556133688453&cacheVersion=1&api=v2) and the accompanying document [Introduction to Variable Cascade in DDI.docx](https://ddi-alliance.atlassian.net/wiki/download/attachments/689242207/Introduction%20to%20Variable%20Cascade%20in%20DDI.docx?version=1&modificationDate=1556134115414&cacheVersion=1&api=v2) both developed at the Train-the-Trainer workshop at Dagstuhl in 2018 gives a good description of this topic, and could be integrated with this portion to provide a useful documentation of some of the basics. Such a document might also wish to address other aspects of DDI 4, such as representations. This documentation will need to be the subject of further work in the MRT working group.

1. This is how Datum is defined in the [DDI4 prototype](https://lion.ddialliance.org/ddiobjects/datum) documentation and [GSIM](https://statswiki.unece.org/display/clickablegsim/Datum).docx [↑](#footnote-ref-1)
2. [Example](https://www.w3.org/TR/vocab-data-cube/)s come from the W3C RDF Data Cube Vocabulary recommendation [↑](#footnote-ref-2)