View points on data points: A shared vocabulary for cross-domain conversations on data and metadata

George Alter[1], Flavio Rizzolo[2], Kathi Schleidt[3]

Abstract

Sharing data across scientific domains is often impeded by differences in the language used to describe data and metadata. We argue that disagreements over the boundary between data and metadata are a common source of confusion. Information appearing as data in one domain may be considered metadata in another domain, a process that we call "semantic transposition." To promote greater understanding, we develop new terminology for describing how data and metadata are structured, and we show how it can be applied to a variety of widely used data formats. Our approach builds upon previous work, such as the Observations and Measurements (ISO 19156) data model. We rely on tools from the Data Documentation Initiative's Cross Domain Integration (DDI-CDI) to illustrate how the same data can be represented in different ways, and how information considered data in one format can become metadata in another format.

Keywords

Metadata, data sharing, data interoperability

Acknowledgments

This paper grew out of the CODATA working group on Semantic Integration, and we are grateful to Simon Cox (chair) and other members of the group for their encouragement. Rob Atkinson, Alejandra Gonzalez-Beltran, Larry Hoyle, Hylke van der Schaaf, Chris Schubert, and John Wieczorek provided helpful comments and guidance on earlier drafts of this paper. This paper would not have been possible without the path-breaking work of the DDI Alliance Cross Domain Integration Working Group. In memoriam to Herbert Schentz, we carry on the good fight towards semantic data interoperability.

Problem Statement

Although the value of sharing data across scientific domains is rapidly increasing, conversations about data are often very difficult. There are many types of scientific data, and each discipline has developed its own standards, procedures, and language about data. Combining data from multiple sources becomes a very frustrating process when common terms, like 'observation' and 'attribute', are used differently across scientific domains. Differences in use of the term "metadata" are especially problematic. We will show that there is no fixed boundary between "data" and "metadata," and that information viewed as data in one discipline may be metadata in another. To achieve the FAIR goal of Interoperability (Wilkinson, et al., 2016), we must overcome not only different ways of structuring data but also different ways of conceptualizing and describing data. This paper defines terms describing fundamental aspects of metadata that can be applied consistently across all disciplines.

Our work builds on and extends the DDI Alliance's Cross Domain Integration (DDI-CDI) model, which describes how data can be arranged in different ways (DDI Alliance, 2020a). DDI-CDI is an important departure from DDI's original focus on describing data arrayed as 'variables' (columns) and 'cases' (rows) (Vardigan, Heus, & Thomas, 2008). DDI-CDI provides a bridge to data structures used in scientific domains that organize data around other concepts, such as 'observations' and 'features.' However, DDI-CDI has little to say about the metadata accompanying each data structure. Consequently, information appearing as data in one data structure may disappear in the transition to a different data structure.

We extend DDI-CDI by applying its constructs to metadata as well as data. Metadata is data too, and we show how DDI-CDI can be applied to data structures containing metadata. In our extended version of DDI-CDI, information is not lost when data are moved from one data structure to another, because we map transitions to and from data and metadata. We illustrate this approach by showing how data organized in "Long" format can be translated into "Wide" and "Multidimensional" formats. In particular, we sketch the path from an individual data-point expressed in accordance with the OGC Observations & Measurements data model (ISO 19156; Cox, 2011) into the tabular representations described by the DDI Codebook format and the multidimensional (n-cube) format described by the SDMX (Statistical Data and Metadata eXchange) standard (Statistical Data and Metadata Exchange (SDMX), 2013, 2021).

We believe that confusion about data restructuring is often due to what we call "semantic transposition." We define **semantic transposition** as

> the relocation of the representation of a characteristic from the data structure to the data content, that retains isomorphic coherence between representations.

This occurs because information about the meaning of a measured value may be either internal or external to a data set. For example, if the measured value is 32, we need to know whether the characteristic being measured is age, temperature, or something else. Some scientific domains are accustomed to including this information in the same data array as the measured values, but others provide a separate "metadata" file that attaches meanings to areas in the data array. Semantic transposition occurs when the data are restructured in a way that moves information into or out of

the data set, shifting information about the characteristic measured by the value from the data array to the descriptive frame. In other words, the boundary between 'data' and 'metadata' is flexible.

Semantic transposition has been discussed in the computer science literature, where it is described as data-metadata translation (Hernández, Papotti, & Tan, 2008; Papotti & Torlone, 2009). A common example is the transformation of stock ticker data, which is an illustration of the "pivot-unpivot" problem. A stock ticker reports data with three columns: time, company identifier, value. For the purposes of analysis, these data are often transformed (pivoted) to a matrix with one row per observation time and values arranged in separate columns for each company. This transformation involves transposing the company identifier from data to metadata (column name). Computer scientists see this as a problem of mapping across database schemas, but in this case the target schema depends upon the content of the data, which cannot be specified in advance (Hernández et al., 2008). Several ways of formalizing and automating data-metadata translation have been proposed (Beine, Hames, Weber, & Cleve, 2014; Britell, Delcambre, & Atzeni, 2016; Wyss & Robertson, 2005; Xue, Shen, Nie, Kou, & Yu, 2013). The pivot-unpivot transformation is common in statistical analysis, where it is called "long to wide," and we link the choice between long versus wide data formats to data cultures found in different scientific domains.

The analysis presented here is particularly important for data stewards serving the social sciences. The standard format of data in the social sciences is the DDI-CDI "wide" data structure, which is a rectangular matrix of columns ("variables") and rows ("observations"). Unlike other data structures, "wide" format does not assign roles to different columns. In particular, there is no way to indicate that one column describes an attribute of another column, such as an indicator of data quality or an estimation method. Social science data repositories have responded to this problem by developing a robust and detailed metadata standard, DDI, which provides a number of ways to annotate important aspects of a "wide" data array. Consequently, metadata plays a much broader role in the social sciences than in other scientific domains, and extending DDI-CDI to trace semantic transposition is important for the interoperability of social science data with data from other domains.

Overview

We proceed in steps intended to bridge practices and understandings in multiple scientific domains.

First, we begin by defining basic concepts. This step is essential, because different domains often use the same words to mean different things. Although we borrow freely from various sources, we offer our own definitions to provide a consistent and comprehensive terminology.

Second, we describe a theory of observation based on the ISO 19156 - Observations and Measurements (O&M) standard. Since O&M was primarily developed from experience in the ecological and earth sciences, we use an example from the social sciences to emphasize its universality.

Third, we examine four common data structures described by DDI-CDI. We begin with a "simple observation" consistent with O&M and show how it can be represented in DDI-CDI constructs. Then, we trace the movement of aspects of an observation as it is transformed from Long to Wide to Multidimensional data structures. We represent each data structure in a tabular format to show how information flows from one data structure to another, even though the data may not be tabular in

practice. This discussion illustrates semantic transposition as defined above, and we use the terms and concepts of DDI-CDI to introduce Variable Definition Data Structures and Dimension Definition Data Structures, which complement and explain each data format.

We also provide an Appendix with machine-actionable descriptions of the steps involved in transforming data from Long to Wide to Multidimensional using Structured Data Transformation Language (SDTL). SDTL is an independent language for describing data transformation commands in standard metadata formats, like Data Documentation Initiative (DDI) and Ecological Metadata Language (EML) (San Gil, Vanderbilt, & Harrington, 2011).

Our conclusion reflects on the difficulty of combining data from different scientific domains and the importance of semantic transposition as a source of misunderstanding.

Basic Concepts

Before we dive into the details of the CDI methodology and apply it to data concepts, we introduce a set of basic terms. We are working across a wide range of scientific domains ranging from environmental to social sciences. Each domain has a unique entrenched terminology that is often at odds with usage in other domains. For example, the "Characteristic" under observation may be referred to as the "Property" or "Variable" in other communities. This section defines the terms used in this document together with the meanings we apply to them in the hope that this clarification helps to avoid subsequent misunderstandings.

Instance Value

We use Instance Value to refer to the smallest atomic unit of data. An Instance Value may be a number, text, boolean, or any other data type. An Instance Value may result from a measurement process, or it may describe the measurement process itself. "Child" and "-10" can be Instance Values measuring age and temperature, but the text strings "Age" and "Temperature" may also be Instance Values.

Characteristic Value

We use Characteristic Value to refer to an Instance Value that contains a measured value describing a single characteristic of a specific Entity. The Entity may be a person, place, thing, total for a region or a year, etc. A Characteristic Value does not explain its own meaning. The value "-10" may refer to a temperature, which could be measured on the Celsius or the Fahrenheit scale, or it may be the difference between a test score and the mean of all scores. "Hazel" may be a person's name or an eye color. Characteristic Values are only meaningful if they are accompanied by additional descriptive information, such as the Characteristic being described and methodological details of the data acquisition process.

Characteristic

Characteristics explain the meaning of a Characteristic Value. "Name" and "eye color" are both Characteristics that could result in an Instance Value of "Hazel." Our understanding of a Characteristic Value must be informed by knowledge of the Characteristic that it describes. Some commonly used terms for this concept are attribute, parameter, variable, observed property (or just property), measurand, analyte. Individual domains become even more specific, with geology field observations utilizing terms such as strike and dip, lithology, alteration state, etc.

Entity

An Entity is a physical, digital, conceptual, or other kind of thing with fixed aspects, that has separate and distinct existence, real or abstract (Provenance Working Group, 2013; ISO/TC 211 Terminology Maintenance Group, 2020).

This broad definition is consistent with emerging usage in communities promoting data documentation and exchange. We prefer these definitions to domain-specific terms like "unit of observation," "statistical unit," and "feature of interest."

An entity may also encapsulate a temporal aspect, such as a moment or period of time. Thus, a person who held three jobs during a calendar year may be modeled as three Entities (pertaining to their employment), each of which existed for a part of the year. Similarly, a household is an Entity composed of persons (individual Entities) who live in a defined space or share a common budget. Thus, Entities may be defined as composites of physical, temporal, and conceptual units.

When data are transformed or re-organized, we often change the Entity that they describe. This is most apparent when data are aggregated, as in one of the examples below. If we count the number of women enumerated in a census, the Characteristic "Number of Women" refers to an Entity defined by the geographic coverage and date of the census.

Time may also be used to create more disaggregated Entities. Suppose that the heights of a group of school children were measured several times. These data can be arranged in a Wide format showing multiple heights for each child, or they can be in a Long format where the Entity is an observation for one child on a specific date. (See below for formal definitions of Wide and Long formats.)

Value Domain

A Value Domain is the set of allowed values for a Characteristic, Qualifier, or Identifier. The Value Domain for temperature in Kelvin, for example, would be all real numbers greater than or equal to 0. Name has a Value Domain that includes "Hazel," "Fred," and "Wilma." The Value Domain of eye color includes "Hazel," "Brown," "Blue," and "Green."

Qualifier

We use the term Qualifier to refer to additional information about a Characteristic Value. Data are often annotated with information about the measurement procedure, the instrument that was used, date and time of measurement, geographic location, verification or validation procedures, etc. Qualifiers are attributes of attributes. Information of this kind is often essential for comparing measurements from different studies or for deciding how much confidence to place in a specific Characteristic Value.

Identifier

Identifiers associate an Instance Value with an Entity or a type of Characteristic or a type of Qualifier. An ID number for a subject or feature is the most common kind of Identifier, but Identifiers may also pertain to the type of a Characteristic or Qualifier.

We may characterize Identifiers by their functions as

- Entity Identifiers

- Characteristic Identifiers
- Qualifier Identifiers

Key

A Key is an Identifier or set of Identifiers that uniquely reference a Characteristic Value. Thus, a person has only one place of birth, and any combination of Instance Values that uniquely associates an individual with a place of birth can be used as a Key. A Key also references any Qualifiers and Identifiers associated with its Characteristic Value.

Keys are Dataset specific, and the number of Identifiers required to compose a Key can change if the structure of the data is modified. For example, if a study collects heights of school children, an identifier for each child can be used as a Key. If the study re-measures the same children at a later date, a Key for the combined data requires both the child's ID and the date of measurement. Thus, a child's ID is not a unique Key when children are measured more than once. Note that the Key points to a unique Characteristic Value (height) and not to a person (child). This example also shows that a Qualifier can be used as an Identifier within a Key. Date of measurement is a Qualifier of height, but date of measurement also serves as an Identifier when it is part of a Key. In other words, when height is measured more than once, date of measurement plays two roles. It is both a Qualifier that affects the interpretation of height and an Identifier that distinguishes among multiple measurements of the same child.

A Key can be associated with more than one Characteristic Value (age, place of birth, date of birth, mother's name), but only one Characteristic Value for each Characteristic can be associated with a specific key.

Procedure

A Procedure is the underlying methodology utilized to ascertain the value of a Characteristic Value. Knowledge of this methodology is often necessary to understand the applicability of the available data to a specific use case.

Datapoint

A Datapoint is a container for an Instance Value, which may be a Characteristic Value, a Characteristic, a Qualifier, or an Identifier. One can think about a Datapoint as a cell in a matrix, such as a spreadsheet. Some cells in the spreadsheet are the Characteristic Values that we plan to study, but other cells contain explanatory information about what Characteristic was measured when, where, and for whom.

Full Simple Observation

We use the term Full Simple Observation to refer to a Characteristic Value with all of its associated Identifiers and Qualifiers. This is the most atomic level of usable data, because it brings together a measured value with information about what was measured and how measurement was performed. The components of a Full Simple Observation may appear in the same place, as in a row of a spreadsheet, or in linked locations, such as tables in a relational database.

Data Set

A Data Set is a collection of Datapoints that have been organized in a known way. The structure of the Data Set tells us which Datapoints are Characteristic Values, Characteristics, Qualifiers, Identifiers, etc. Thus, the format of a Data Set sets our expectations about each of its Datapoints.

Data Structure

A Data Structure describes the roles played by various Datapoints in a Data Set. The Data Structure indicates which Datapoints are Characteristic Values, Characteristics, Qualifiers, and Identifiers. A "Logical" Data Structure describes the roles played by the Datapoints in a data set. A "Physical" Data Structure shows how Datapoints are formatted into rows, columns, and files. A Logical Data Structure may be instantiated in more than one type of Physical Data Structure.

Some Physical Data Structures do not include all of the information required to make a Data Set usable. For example, data formatted as Comma Separated Values (CSV) without semantically meaningful column headers is useless without accompanying documentation of the Characteristic and role played by each column of Datapoints. If this documentation is machine actionable, it can be described as a Data Structure in its own right.

Metadata

The preceding discussion avoided using the word "metadata." One could say that Instance Values are data while Characteristics, Qualifiers, and Identifiers are metadata, but we do not consider statements of that kind helpful. As we mentioned above and will illustrate below, Characteristics can be Datapoints within a Data Structure or they can be supplied elsewhere. If "temperature" and "name" are included in a Data Structure, they can be processed like any other Datapoint. For example, one can count the number of Characteristics in a dataset. In other words, the difference between "data" and "metadata" depends upon how a Datapoint is used and not on how it is provided.

We argue here that the assignment of content to "data" versus "metadata" is arbitrary. Most scientific domains are accustomed to Data Structures that specify which concepts are embedded in the **data content** and which concepts are part of the **descriptive frame**. These decisions are often motivated by technical considerations about types of data and modes of analysis, as well as user perspectives on the usage of the data, but the same data can be represented in alternative Data Structures. Reshaping data into a different Data Structure is an act of semantic transposition that determines which content will be provided in the data content and which will go into the metadata (descriptive frame). For example, the Characteristic measured in a Datapoint may be provided in the data or associated with a column name that points to a description of the Characteristic in the metadata.

Cross Domain Integration from DDI Alliance (DDI-CDI)

The Data Continuum

Different use cases entail the use of different structures for data representation. In some cases, precise metainformation detailing the data acquisition process is essential to understanding the applicability of the provided data to the task at hand. In other cases, simplified structures can be of great benefit, reducing the resources required for data provision, transport and use. A modern data provision landscape should encompass both aspects, while ensuring a degree of continuity between these alternative viewpoints on the same data source.

While each of the various data structures utilized is consistent in itself, issues arise when data is transformed from one format to the other, as often required to support a wide array of use cases. It becomes difficult to maintain semantic coherence across structures for those cases when it proves necessary to drill down into the details of the data. In order to expose data in different structures, with differing depths of information contained, it would be advantageous to be able to provide links between parallel concepts, allowing a user to traverse between these different structures. DDI-CDI offers a way to encapsulate the underlying essence of the data being provided. In the following sections, we will describe the relevant concepts from this emerging data alignment model.

We focus on two ways that DDI-CDI helps us to characterize Data Structures. First, a Data Structure can be defined by its Keys. As we discussed above, a Key is a set of one or more Identifiers that point to a Characteristic Value. Data Structures differ in the number and types of Identifiers, i.e., Keys, required to uniquely identify a Characteristic Value. Second, DDI-CDI describes the roles that Instance Values play in different Data Structures. The role played by an Instance Value may differ across Data Structures. For example, an Instance Value that is part of a Key in one Data Structure may not be part of a Key in another Data Structure. DDI-CDI refers to roles as "Components", which will be described below.

We differ from DDI-CDI in several ways.

- We use the same concepts to describe the data content as the descriptive frame, i.e., metadata. We emphasize that metadata is also data. Transferring data from one Data Structure to another often involves moving information (Instance Values) from the data section (data content) to a metadata section (descriptive frame), i.e., semantic transposition.
- DDI-CDI does not specify relationships between Qualifiers and Characteristic Values. We consider this relationship essential for understanding differences among data structures.
- Although we use DDI-CDI concepts to describe logical data structures, we also provide examples showing simplified physical data structures. We hope that these examples will help readers to see beyond an abstract discussion of concepts to practical applications.
- We introduce a data structure called "Nested Name-Value Pairs," which is not included in the DDI-CDI. "Nested Name-Value Pairs" is similar to the "Key-Value" data structure included in DDI-CDI, but name-value pairs may be nested, which is not possible with "Key-Value" pairs. We consider "Nested Name-Value Pairs" a reasonable extension of DDI-CDI, and we hope that it will be added to the DDI-CDI specification.

Theory of Observation

What is an Observation?

Most information we have about our surroundings can be seen to be the outcome of observations or measurements upon our world. The essential characteristics of an observation have been elaborated within the standard ISO 19156 - Observations and Measurements (O&M) (ISO 19156), leading to a richly structured model as follows.

The essence of an observation is a relation assigning a value (the range of the observation relation) to an Entity (the domain of the observation relation). In addition, various additional pieces of observational metainformation are linked to this relation via the observation object, including:

- The property or Characteristic of the Entity for which a value is being provided, e.g. color, temperature. In a simplified structure, this property would be the name of the relation between the Entity and the value;
- The Procedure used in obtaining the value for the Entity. This can be essential for interpreting the value provided, as different methodologies can deliver vastly different results in dependence on external factors;
- Temporal information pertaining to the observation, specifically the phenomenon time and the result time;
- Spatial information on where the Entity being observed was located at the time of observation.

In a similar vein, information such as the measurement device utilized, the person performing the measurement or the facility in which this act took place is often provided, as well as references to other observations providing essential contextual information are foreseen within this model, but omitted here for brevity.

In the Figure 1 below, we show the conceptual model underlying the update of O&M, to be released as ISO 19156:2022. The core of the Observation consists of 2 associations to the left: Domain and Range; Domain associates the Observation with the feature-of-interest, the Entity upon which the Observation provides a value for a Characteristic, while the Range associates the Observation with the actual value for this Characteristic pertaining to the feature-of-interest. The ObservableProperty provides the Characteristic under investigation, while the ObservingProcedure describes the measurement methodology. In addition, information on the Observer, e.g., a sensor or human providing the value of the Characteristic, the Host, e.g., the facility the sensor mounted at, as well as deployment information linking an Observer to a Host can be provided.

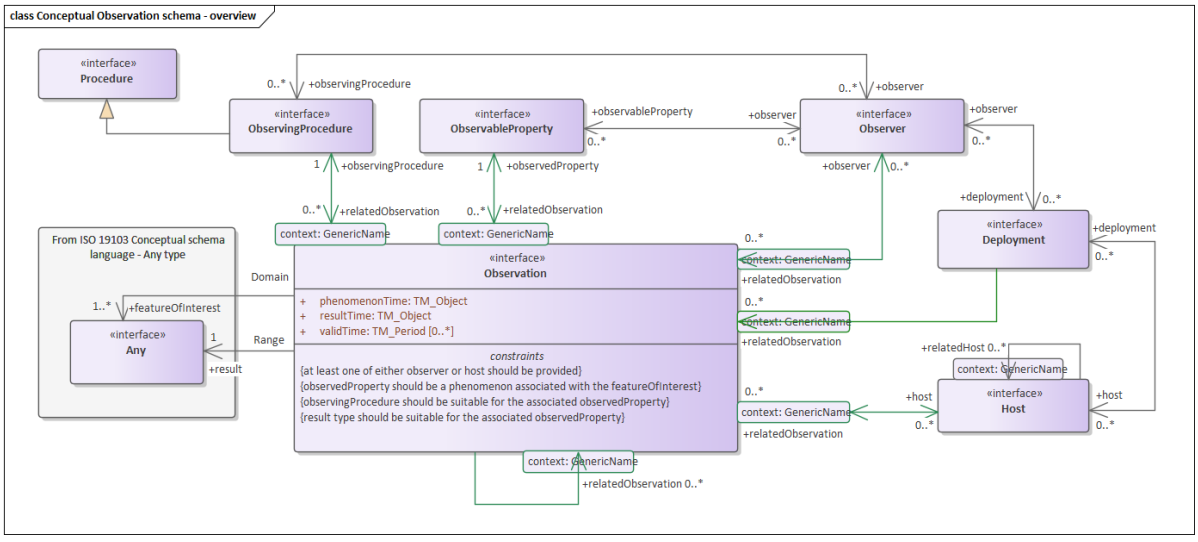


Figure 1. Class Diagram for Observation and Measurement Model

This leads to a precise but complex representation of all aspects of the observational process deemed relevant to later interpretation and use of the data. While access to these details may be essential to understanding the applicability of the data, when it comes to further processing steps, this excess baggage proves cumbersome; simpler formats are required.

An Example Relational Observation

In the following example, we will concern ourselves with the gender of an individual. In the simplest representation, we could expect an Entity of type Person to have an attribute or operation gender, providing this characteristic as a string value, ideally referencing a standardized vocabulary. In Figure 2 below, we have modeled this example as a UML interface Person with the operation gender() of type GenderValue (a data type providing a string value representing the gender of the individual); an instance Simple1001 has been derived from this interface, and gender provided as a reference to a URI representing the value "Female".

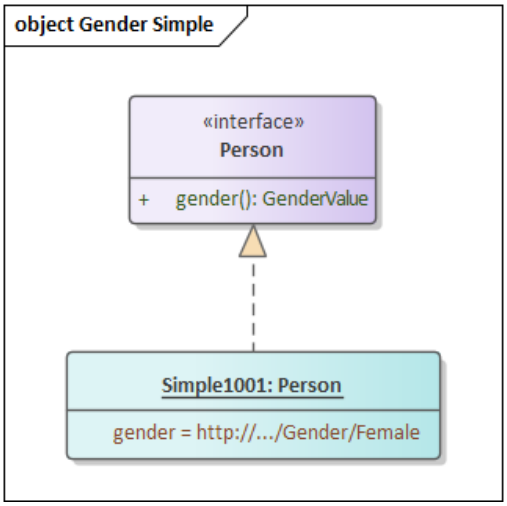


Figure 2. Simple Instance Diagram

Suppose that we want to add information about how gender was determined. The concept of gender can be reified from an attribute to a class, which can have more than one attribute. (See Olivé (2007, Chapter 6) for a definition of reification.) In Figure 3, we show the GenderDetermination class with two attributes ObservingProcedure and Gender Value. Reification of the attribute "Gender" to the class GenderDetermination allows us to show that a specific measurement procedure, "External_observation," applies only to the determination of gender and the resulting value "Female."

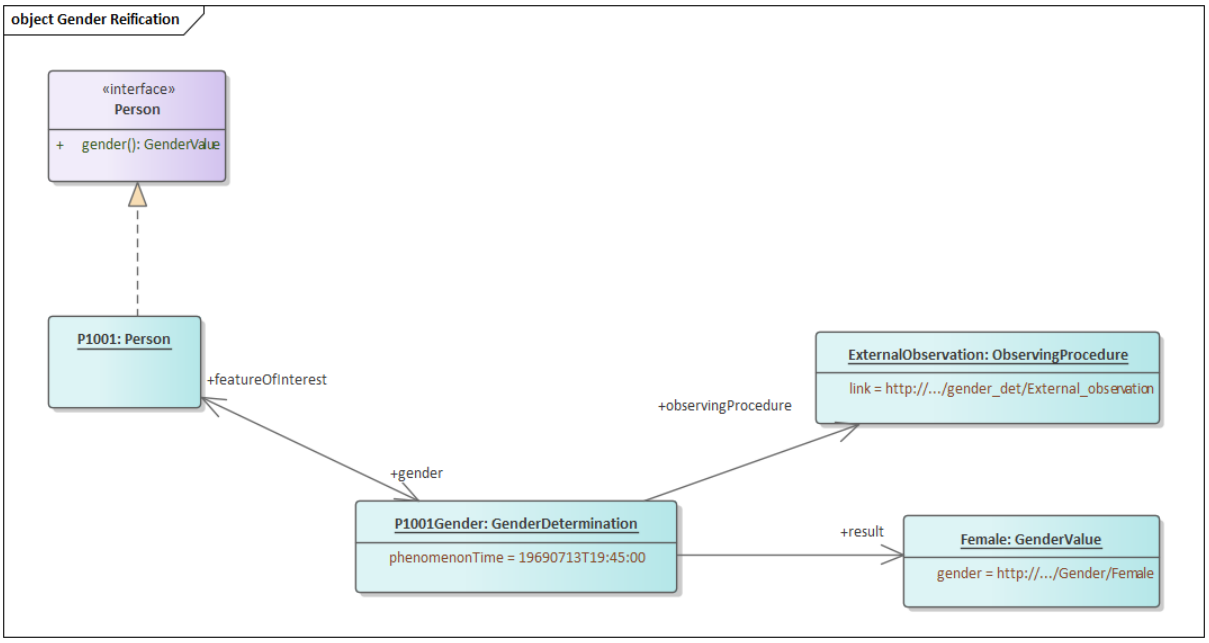


Figure 3. Reified Instance Diagram

Rather than creating a dedicated class for every attribute, O&M further abstracts the reified class to the concept of Observation in which the characteristic being represented is provided via the ObservableProperty association and class. In Figure 4 below, we show the full O&M representation of our gender example, where explicit interfaces are provided for the relevant observational metainformation concepts. Note the semantic transposition of the denotation of the measurement characteristic “Gender” from the name of an attribute on Person to the content of the name of the ObservableProperty; we will repeatedly observe this semantic transposition or flip-flop between data content and data structure for the provision of the characteristic under investigation as we further analyze the various structures commonly used for the representation of observational data.

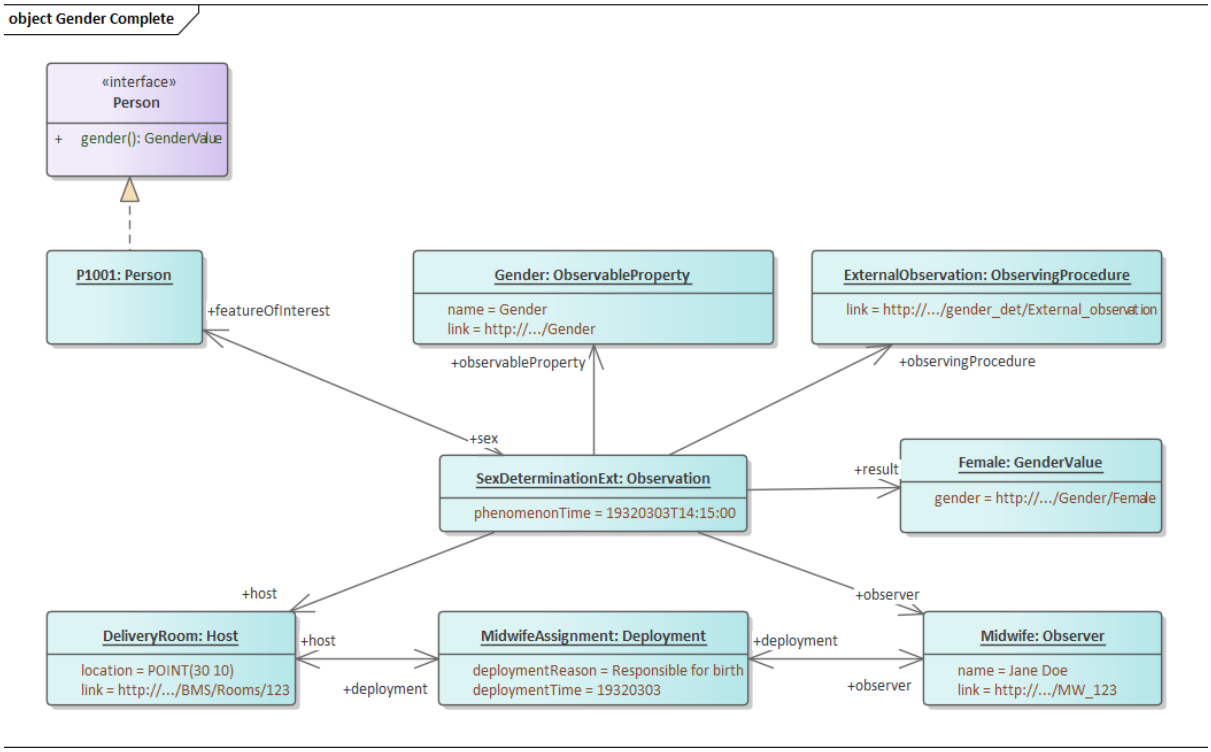


Figure 4. Instance Diagram with Identifiers and Qualifiers

In order to bridge the gap between a relational data store and existing external vocabularies, most interfaces foresee a link attribute by which a reference to the corresponding concept can be stored. Alternatively, the link can also be utilized to reference any external source providing additional information on this Entity.

We are aware that gender determination is a sensitive topic, but the changing understanding of gender illustrates our point. When sex was considered an immutable biological characteristic, all measurement procedures were expected to yield the same result. As we recognize the right of individuals to determine gender for themselves, the procedure used to ascertain gender becomes more important. We do not expect everyone to self-identify with the biological sex ascribed to them at birth or to be limited to the binary choice between female and male. Measurement procedures have important consequences.

The true power of such a richly structured representation becomes clear when we add additional observations on the gender of this individual over time. While the Observation shown above describes the gender determination made at birth, additional determinations could be made over the individual's lifetime, following various methodologies. Continuing this example, we now add a gender self-determination observation in Figure 5. As this observation is performed by the subject, there is no need to provide information on the Observer and Host; the following diagram illustrates this observation.

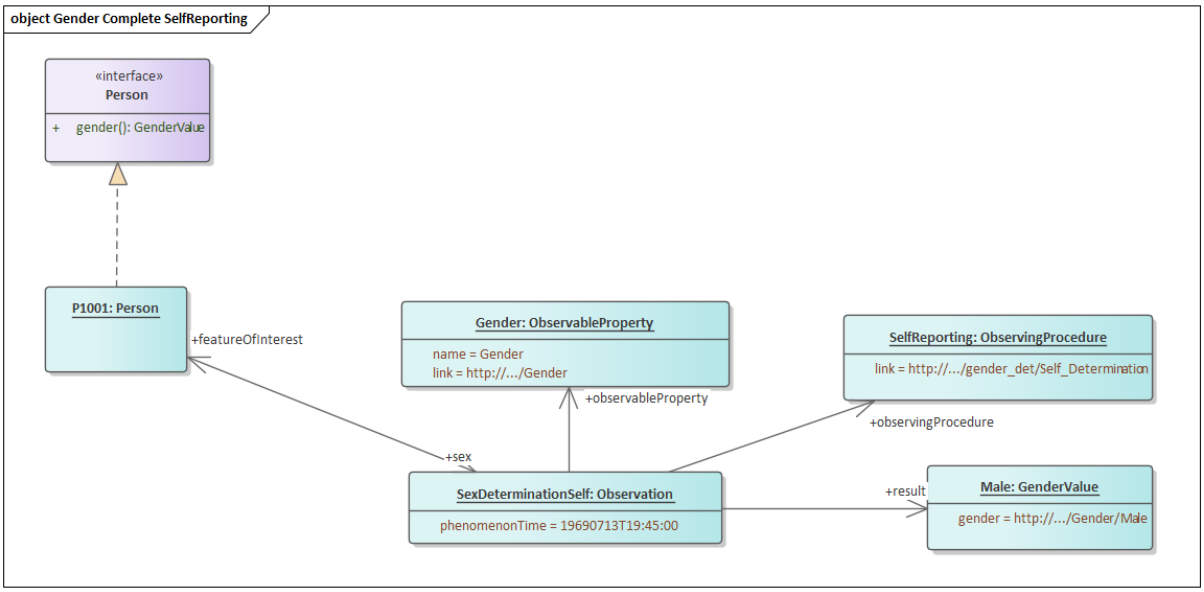


Figure 5. Instance Diagram with Identifiers and Qualifiers for First Alternative Procedure

A final gender determination is performed after the death of the individual in an attempt to clarify the contradictory gender markers available (Figure 6). For simplicity we have defined the Observer as the coroner responsible for this step, while in a real-world system, this object should probably represent the DNA Analysis equipment.

This example illustrates the issues encountered when such information is oversimplified. If we only consider the information pertaining to gender being exposed via the simple interface, there are two options for provision of the data, neither proving satisfactory:

- The gender attribute changes over time, thus returning different values for the same individual for three time-instances as follows:
    - 19320303T14:15:00: Female
    - 19690713T19:45:00: Male
    - 20051203T08:30:00: Female
- An undetermined value for gender of the individual

The type of representation required depends on the actual use case. When looking for suitable data, the complex richly structured representation provided by the O&M standard is often essential to allow a domain expert to determine if the data is fit for purpose, but once the data has been vetted and deemed appropriate, simpler representations allow for more efficient data transfer and portrayal.

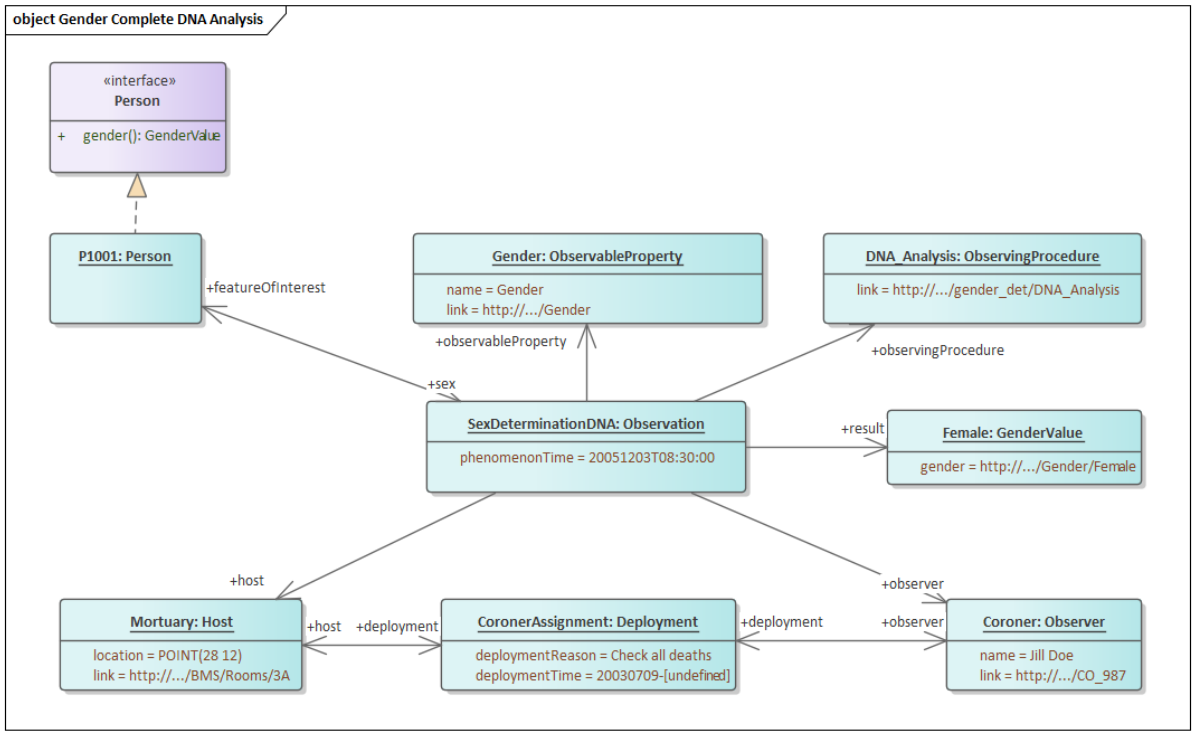


Figure 6. Instance Diagram with Identifiers and Qualifiers for Second Alternative Procedure

Common Data Structures

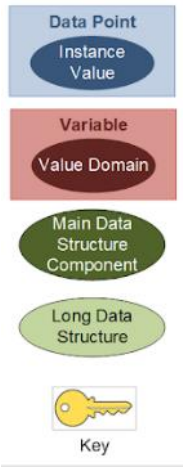
Conventions and Data Used in this Document

Data takes many forms in its transformation from primary microdata to highly aggregated data stores. During this process, depending on the type of structure utilized, individual concepts can shift from being part of the content section to the descriptive frame. In this section, we describe the different structures involved at the different levels of this process before going into the details of tracing the individual concepts throughout this process.

In order to illustrate this process, we use a simple example dataset, providing data pertaining to the following characteristics on two individuals:

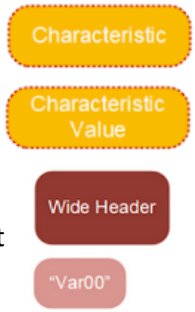
- Name
- Gender
- Born
- Died
- RefArea

We use DDI-CDI concepts to represent the Logical Data Structure of a Relational Observation in Figure 7. For ease of representation within this paper, we have utilized the following graphical representation of the DDI-CDI concepts:



- **Data Points**, represented as blue boxes, with the **Instance Value** provided in the darker blue oval contained
- **Variables**, represented as pink boxes, with the **Value Domain** provided in the darker red oval contained
- Roles of elements within a structure, also referred to as **Data Structure Components,** are represented as green ovals.
    - Dark green ovals are used for elements that play the same role in every structure.
    - Light green ovals are used for elements playing a role that is specific to a particular format.
- **Keys**, represented by golden key symbols, indicating which concepts must be combined to form the key for a specific observation

Additional concepts we have defined for this paper



- Golden rounded boxes are used to indicate **Characteristics** and **Characteristic Values**. These are terms defined in this paper showing continuities across data structures that can be lost in the terminology differences between scientific domains.
- Pink and red rounded boxes introduced in Figure 11 are used to represent features of data arrays, such as column headers and variable names, that are part of the descriptive frame.

Named arrow representations:

- “identifies” (in blue): indicates the Data Points that are uniquely identified by the given Key.
- “identifies” (in green): indicates the Variable a given Instance Value references in a Variable Descriptor Component.
- “has" (in green): indicates the Data Structure Component that is part of a given Data Structure.
- “is defined by”: indicates the Variable that gives meaning to a given Data Structure Component.
- “has value from”: indicates the Value Domain from which an Instance Value is taken.
- “name”: indicates the name of a Data Structure Component.
- “refers to”: indicates the Reference Value a Descriptor points to.

Unnamed arrow representations:

- yellow lines show aggregation, indicating the Instance Value that is part of a given Key.

- dashed associations (in red): terminology mapping that relates the notions of Characteristics and Characteristic Values to Variables and Instance Values, respectively.
- aggregation (in red): indicates a Data Structure Component name is part of a Header.

Relational Representation of Observations

We start with a richly structured relational representation of observations, following the O&M model previously described. In Figure 7, we show the determination of the characteristic Gender utilizing the procedure DNA Analysis to determine that the individual with identifier 1001 has the value Female. Concepts such as Observer and Host have been omitted for brevity.

This example includes only one Characteristic Value, "Female", which is associated with the Characteristic "Gender".

In Figure 7, the Instance Value "Female" of Variable "Gender" is a Measure Component. "Gender" appears as an Instance Value that belongs to a Value Domain together with other Characteristics, such as age, height, hair color, place of birth, etc. DDI-CDI calls this role a Variable Descriptor Component. In other words, a Variable Descriptor identifies the Characteristic measured by a Variable. In this example, the Variable Descriptor assigns a name to the variable, but it could also provide a URI referencing a variable description in an ontology.

The variables on the right side of Figure 7 are Qualifiers, which are called Attribute Components in DDI-CDI. Procedure and Time tell us important things about the Characteristic Value ("Female"): how it was determined and when it was measured.

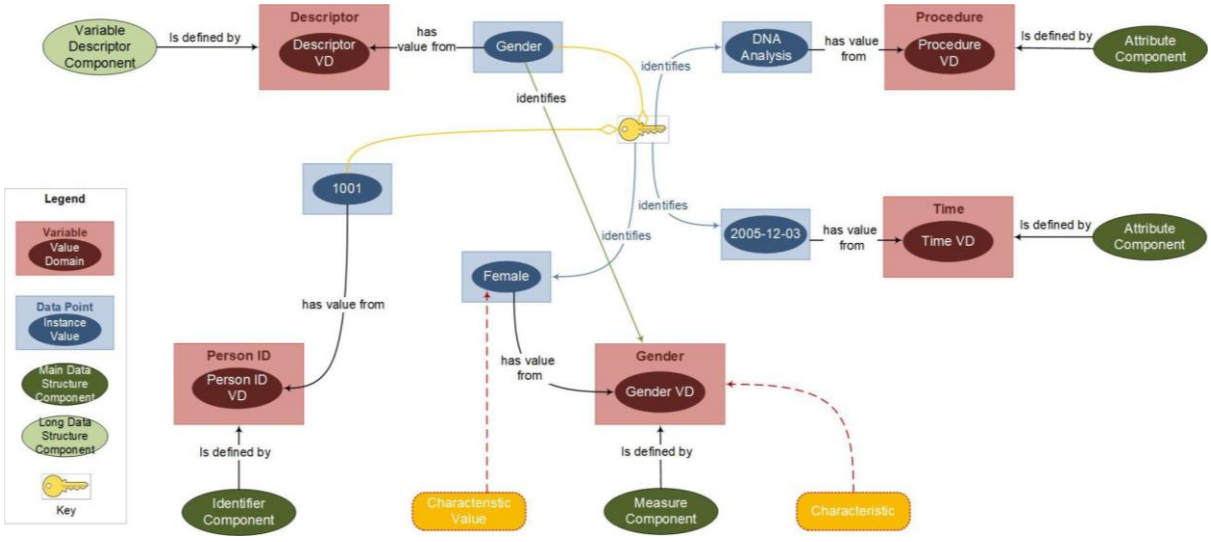


Figure 7. Logical Data Structure of a Simple Observation (one procedure and one time per measure)

The Key for this observation includes two Instance Values "1001", which is an Identifier for the person, and "Gender" which is the Characteristic. The Variable Descriptor is part of the Key, because we may have additional Observations on other Characteristics. The Instance Value of Time is not part of the Key in Figure 7, because we are representing an observation that only occurred once. If there are multiple observations on the same Person at different times using different methodologies, as shown

in our initial Observation description above, Time and Procedure are included in the Key. In Figure 8, the yellow aggregation lines between the key symbol and the Instance Values for Time and Procedure indicate the composition of the Key.

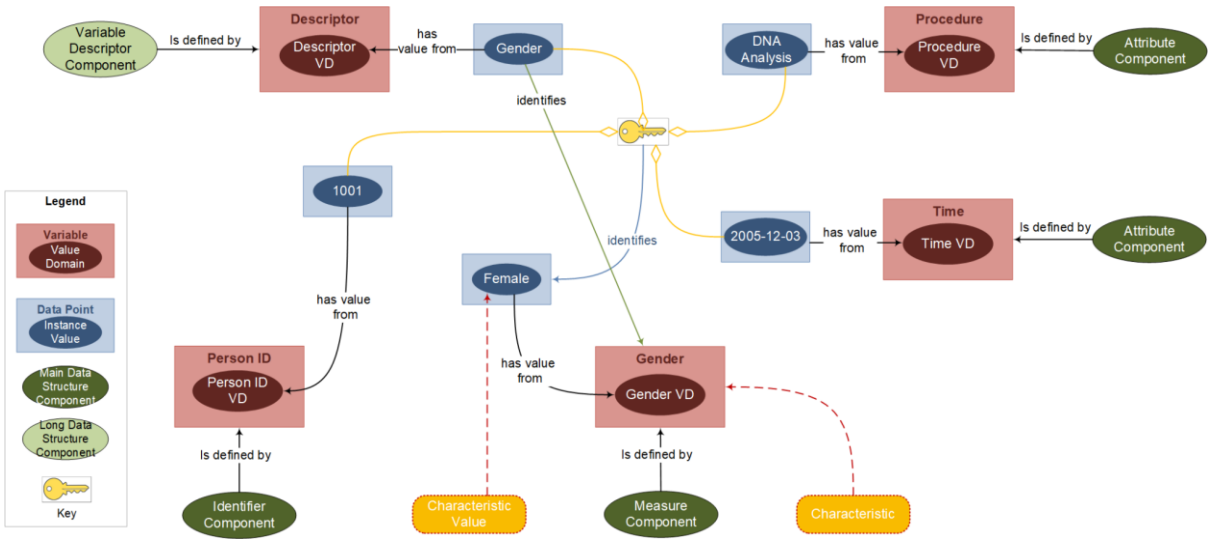


Figure 8. Logical Data Structure of a Simple Observation (Key supporting multiple procedures and times per characteristic)

Tracing Data Through Alternative Physical Data Structures

To illustrate these concepts, we show how the same data are represented in three different Data Structures: Long, Wide, and Multidimensional. DDI-CDI characterizes the data content in each of these formats, but we will show that these models are incomplete without also explicitly exposing the descriptive frame (metadata). In each step from Long to Wide to Multidimensional, information moves from the data content to the descriptive frame. In other words, transposing data to a different Data Structure results in semantic transposition as well. Recognizing that metadata are also data, we can use the tools provided by DDI-CDI to create Variable Description Data Structures for the descriptive frames associated with each Data Structure. The direction of this transposition can go both ways depending on the restructuring; in some cases, information moves back from the descriptive frame to the data content. Thus, our models retain descriptive information that appears to disappear (or reappear) if we focus only on the data content.

The examples below are presented in tabular form, because it is simple to display in print and easy to understand. Tabular formats (e.g., CSV, spreadsheets) are widely used due to their flexibility and simple ingestion by a wide range of analysis tools. However, the same Logical Data Structures could be implemented in normalized relational databases, JSON, RDF, or other physical formats.

Long Format

Long Format, also referred to as narrow or stacked data, or in its most primal form as entity–attribute–value model (EAV) data, is most closely related to relational observations as described in the section above. For each Characteristic Value, there is one row in the table.

In the purest EAV format, the data consists simply of triples with the following structure:

- Entity: The person, object, or thing that is the target of the value provided
- Attribute: The Characteristic (also called variable or property) that is being described by the value
- Value: The Characteristic Value assigned to the entity

Table A provides an example of data encoded in EAV format, in which the Entity is PersonID and the Attribute is called Property.

PersonID	Property	Value
1001	Name	Abigail
1001	Gender	Female
1001	Born	03.03.1932
1001	Died	01.12.2009
1001	RefArea	Newport
1011	Name	Benjamin

Table A: Example Long Format: Entity–Attribute–Value model

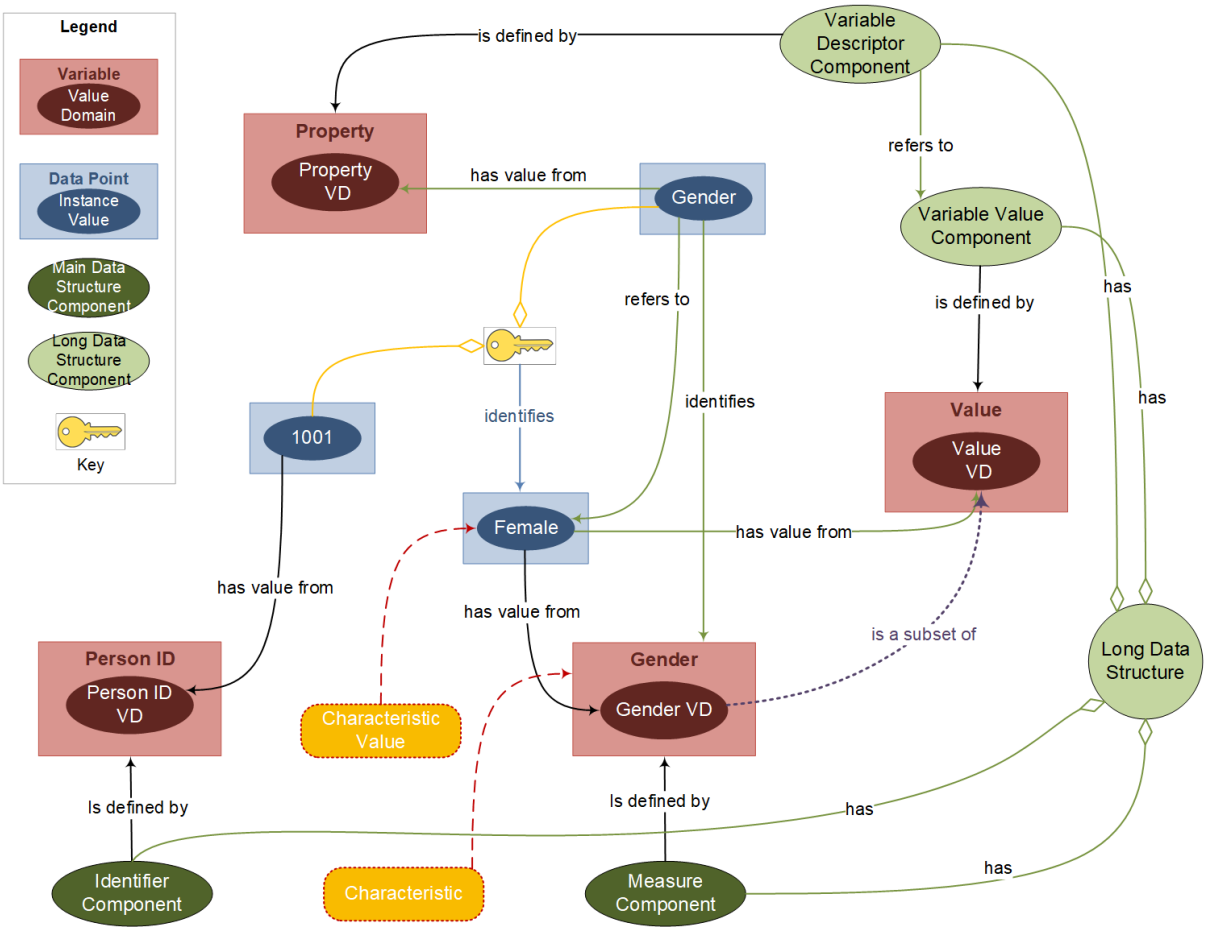


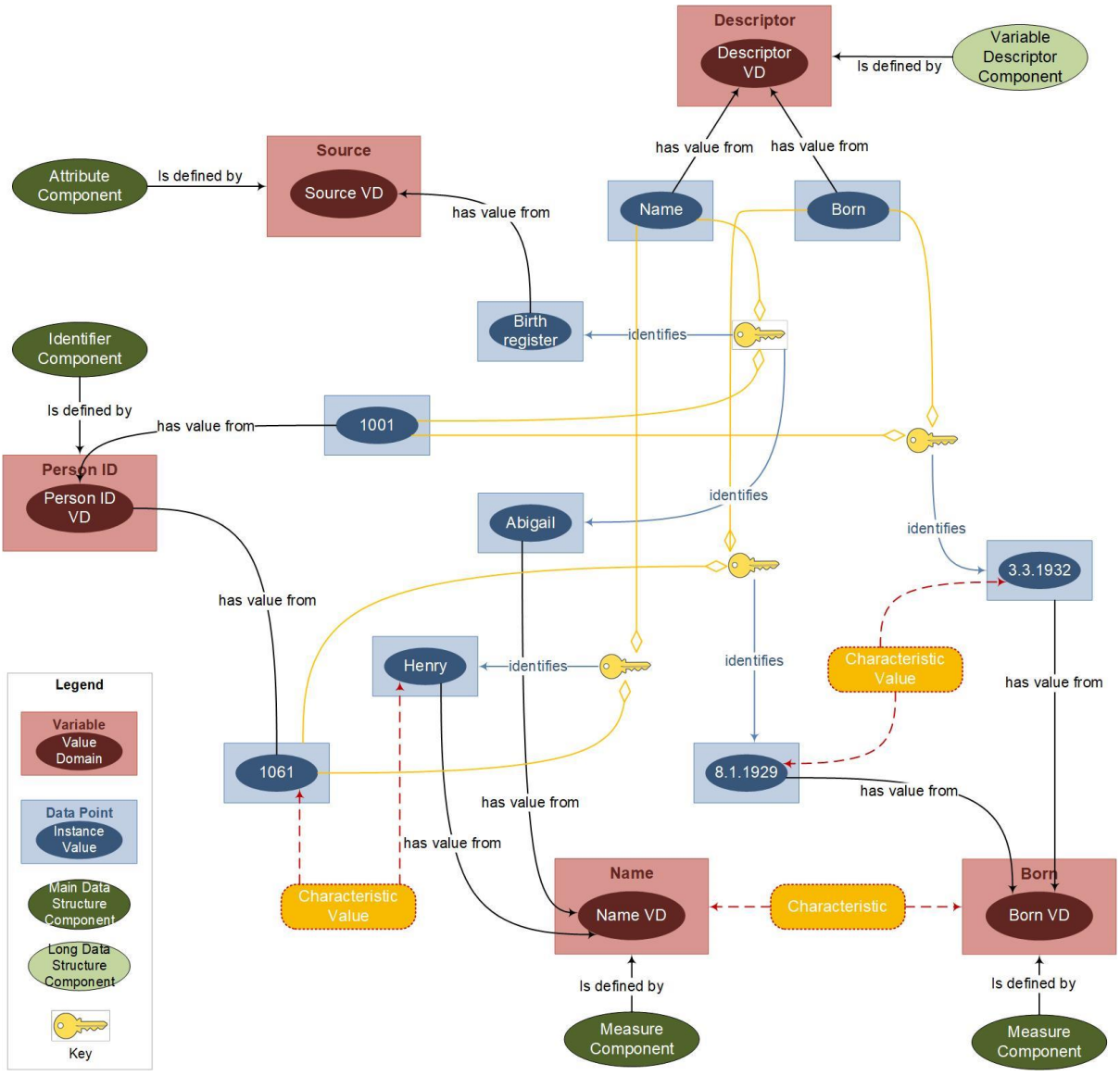
Figure 9. Logical Data Structure of Long Format: Entity-Attribute-Value

Figure 9 shows the Logical Data Structure of a slice of the data in Table A. Figure 9 includes one Measure Component (Gender) and one Identifier Component (Person ID). The Measure Component must be linked to two keys: Person ID and Property, which is a Variable Descriptor Component showing the Characteristic measured in the Characteristic Value. The value "Female" is linked to two value domains. On one hand, "Female" is drawn from the value domain of Gender. On the other hand, "Female" is drawn from the value domain of the EAV column "Value", which is the union of all value domains of attributes in the data including Gender.

Long format can also be extended to provide additional information qualifying each Characteristic Value. A simple example is the provision of source information for each Datapoint, which was included as a Qualifier in Figure 6, as shown in Table B.

PersonID	Property	Value	Source
1001	Name	Abigail	Birth register
1001	Gender	Female	DNA analysis
1001	Born	03.03.1932	Birth register
1001	Died	01.12.2009	Kin report
1001	RefArea	Newport	Drivers license
1011	Name	Benjamin	Birth register

Table B: Example Long Format – Source



 Alter, George; Rizzolo,Flavio; and Schleidt, Kathi (2023) View points on data points, IASSIST Quarterly 47(1), pp. 1-39. DOI: https://doi.org/10.29173/iq1051

Figure 10. Long Format Extended to Include a Qualifier: Verification

Figure 10 extends Figure 9 in two ways. We show two Characteristics (Name and Born) for two observations (1011 and 1061), and we have added an Attribute Component (Source). The Attribute Component in Figure 10 is linked to a Measure Component by sharing the same two Keys: "1011, Name". This means that its Instance Value (Birth register) applies to the Name variable of person 1011. We include only one Attribute Component to avoid further complications in a busy diagram, but we could add Attribute Components with the procedures used to ascertain each of the other three Instance Values: "1011, Born"; "1061, Name"; "1061, Born".

Long Format can be further extended to include the full breadth of information in an observation in a lossless manner. Table C adds columns for two more Attribute Components depicted in Figure 6: Time and Source. Note that these additional columns are both Qualifiers that modify the Characteristic Value (i.e., "Female"), and they are linked to the Characteristic Value by a two-part Key, "1001, Gender".

PersonID	Property	Value	Time	Source
1001	Gender	Female	5.12.2009	DNA analysis[4]

Table C: Example Long Format - Full Simple Observation

Wide Format

Wide (or unstacked) data is structured with a separate column for each Characteristic, such that each row contains all of the Datapoints pertaining to one observed entity. Table D exactly corresponds to Table A, but the Datapoints are arranged horizontally rather than vertically. Consequently, Characteristic Values in Table D are linked to only one Key, PersonID.

PersonID	Gender	Name	RefArea	Born	Died
1001	Female	Abigail	Newport	03.03.1932	01.12.2009
1011	Male	Benjamin	Cardiff	01.08.1929	02.06.2006

Table D: Example Wide Format - Simple representation

Table D poses a problem that was implicit in our discussion of Long Format, but critical for understanding Wide Format. What is the special status of the first line in Table D? We can make this question more conspicuous by re-writing the same data as Table E. Column headings like "Var01" have no intrinsic meaning, and we could just as easily write the same data matrix without any headings at all. Clearly, if Table E is not accompanied by additional information, it is unusable. One might infer the meaning of Var01 and Var02, but it is impossible to interpret the other columns.

Var00	Var01	Var02	Var03	Var04	Var05
1001	Female	Abigail	Newport	03.03.1932	01.12.2009
1011	Male	Benjamin	Cardiff	01.08.1929	02.06.2006

Table E: Example Wide Format - Arbitrary Column Headings

Table E is only meaningful if it is accompanied by Table F. Table F is sometimes described as a codebook or a variable inventory, and it is often distributed as a text file or a spreadsheet. For our purposes, Table F is a simplified version of a metadata file. Scientific domains that distribute data in formats like Table E have developed more elaborate standards for providing metadata in machine actionable formats, like XML, JSON-LD, and RDF. Data repositories serving the social sciences rely on metadata in one of the Data Documentation Initiative (DDI) standards, and repositories serving the ecological sciences use Ecological Metadata Language (EML) among other standards. Our point is that Table E requires Table F to supply information contained in the "Property" column of Table A.

Variable Name	Variable Label	Variable Description
Var00	PersonID	Person Identification Number
Var01	Gender	Gender
Var02	Name	First name
Var03	RefArea	Location of principal residence
Var04	Born	Date of birth
Var05	Died	Date of death

Table F. Variable Descriptions

In Figure 11 we provide Logical Data Structures for both Tables E and F. The left side of Figure 11 shows the data content. As we noted above, Wide Format includes only one Key (PersonID); when we provide other Characteristic Values (Name, RefArea, Born, Died), they are all linked to the same Key. The right side of Figure 11 is a Variable Description Data Structure, that attaches meanings to the arbitrary variable names in Table F. Notice that the Variable Description Data Structure on the right side of Figure 11 has essentially the same structure as the Wide Data Structure on the left side of the diagram. Both structures use a key to reference a Characteristic of an entity. In the Wide Data Structure on the left, the entities are persons who have measured Characteristics, such as Gender and Age. Entities in the Variable Description Data Structure on the right are variables, which have labels, descriptions, and other attributes.

In Figure 11, variable descriptions are linked to data in the Wide Data Structure through their variable names (Var00, Var01), which appear in the header of Table E. Headers may or may not be stored with the data array. For example, column names may be provided in the first row of a CSV file or in a separate document, such as a codebook or DDI XML file. We show the header elements of a data structure in pink and red to indicate its ambivalent position. Although we did not show a header in our discussion of Long Format, we return to this issue below.

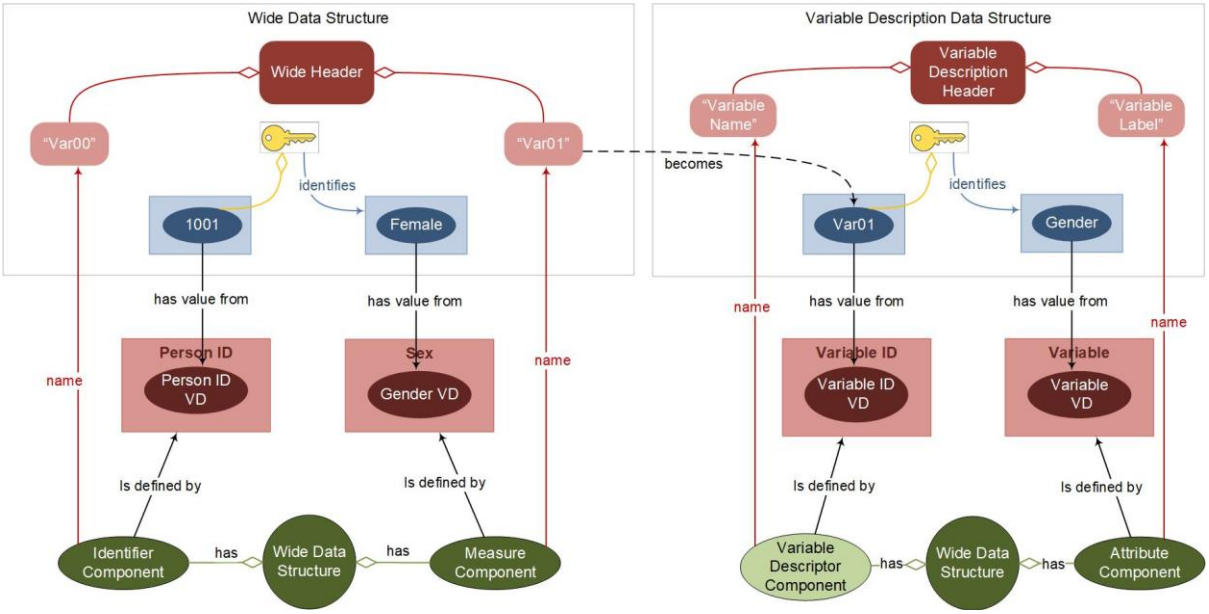


Figure 11. Wide Data Structure with Variable Description Data Structure

When we consider the Wide Data Structure on the left side of Figure 11 by itself, "Var01" is a Measure Component. However, when Figure 11 is considered as a whole, "Var01" is a bridge to information on the right side of Figure 11. In the Variable Description Data Structure "Var01" is a PropertyID, which is used to identify the characteristics of a variable. "Var01" is a Key referencing the Property "Gender", which is the meaning of "Var01" in the Wide Data Structure. In DDI-CDI terminology, Property ID is a Variable Descriptor Component and Property is an Attribute Component. These components are present in Long Format shown in Figure 7, but they are not part of a Wide Data Structure. **This means that Figure 11 taken as a whole has all of the components found in Figure 7 for Long Format above.**

Thus, Figure 11 provides another way of answering the question posed above: What is the status of the first row in Table D? The first row in Table D is a set of variable descriptions. Unlike Table A, these descriptions are not included in the data array itself. Rather, they belong to a separate data/metadata array that must be provided to make the data in Table D meaningful.

The importance of breaking down the distinction between data and metadata is clear if we consider going from Table E to Table A. Even though the measured values are the same in both tables, the column headings in Table E are arbitrary and do not identify the contents of each column (i.e., the Characteristics), which is done in the descriptive frame (Table F). In contrast, Characteristics are given in the data content in Table A. The only way to fill the "Property" column in Table A is to refer to a separate "metadata" table such as Table F. Thus, transposing data from Tables E and F to Table A

requires moving Instance Values from the descriptive frame to the data. To automate data integration across disciplines, the "I" in FAIR, metadata must be provided in standard machine-actionable formats.

Adding Qualifiers to Wide Format

Unlike Long Format, Wide Format does not provide an easy way to associate Qualifiers with Characteristic Values. Since Long Format uses two Keys, the entity Identifier and the Characteristic, Qualifiers are unambiguously linked to the Characteristic Values that they describe. In Wide Format, Qualifiers must be linked to Characteristic Values through their Variable Names. This can be accomplished in a Variable Description Data Structure (metadata) or by including a descriptor for the Characteristic in the variable name, as we show in Table G. In such cases, care must be taken to assure that the metaformat defined for these qualifiers (e.g., the Characteristic name plus "_source" in Table G) is explained to data users.

PersonID	Name	Name _source	Gender	Gender_ source	RefArea	RefArea_ source	Born	Born _source	Died	Died _source
1001	Abigail	Birth register	Female	DNA analysis	Newport	Drivers license	03.03.1932	Birth register	01.12.2005	Kin report
1011	Benjamin	Birth register	Male	Self-report	Cardiff	Drivers license	01.08.1929	Self-report	02.06.2006	Death register

Table G: Example Wide Format - Enriched representation - Full Complex Observation

Multidimensional Formats

Multidimensional data structures, also known as data cubes and multi-indexes, are often used to organize and view large data sets. The axes in a Multidimensional data structure are properties (Characteristics) of groups of observations, and a specific observation is identified by specifying the intersection of a set of dimensions. Gross national product, for example, may be indexed by nation and year. Multidimensional data structures are often used by official statistical agencies for measures obtained by aggregating over persons, households, businesses, or other units of observation. For example, the average number of persons per household may be indexed by region, urban/rural residence, and categories of household income. However, Multidimensional format is also used for non-aggregated data, such as environmental values that can be located in space and time. For example, sea water temperature may be indexed by longitude, latitude and date. Some air quality components offered by the Copernicus Atmosphere Monitoring Service add a vertical component in addition to longitude, latitude and date, providing values calculated from multiband satellite imagery. In this paper, we focus only on those multidimensional data structures providing aggregated data.

We extend our example by showing how the individual-level data in Table G can be converted from Wide to Multidimensional format through aggregation. Table H provides an example in which persons in a Wide data structure, like Table E, are counted by age and gender. The construction of Table H from Table E involves several data transformation steps before aggregation. Since dimensions must consist of mutually exclusive categories, properties with continuous value ranges must be transformed into related properties with discrete values. We have recoded Age into two categories, Young and Old. We provide standardized syntax for describing data transformations based on Structured Data

Transformation Language in the Appendix. In contrast to the transformation from Long to Wide formats above, aggregation inherently results in the loss of information, and extraction of the primary data is no longer possible.

		Age	
		Young	**Old**
Gender	**Male**	3	7
	Female	6	4

Table H: Example Multidimensional Format: Number of Persons by Age and Gender

When data from Table A or E are converted to Table H, we perform a semantic transposition that is independent of the process of aggregation. Consider the Characteristic Value "6" in the southwest corner of Table H. The Characteristic measured as "6" is the number of young, female persons in the set of observations covered by Table H. We know this from the title of Table H, "...Number of Persons by Age and Gender," not from anything in the table itself. In the Long and Wide formats "Female" was a Characteristic Value, but here it is a location on the dimension Gender. Notice that Table H has labels for rows as well as columns and that each dimension has two levels of labels, a dimension name (Age) and a category name (Young). We will refer to "Female" and "Young" as Facets to distinguish them from Characteristic Values. Since labels are part of the descriptive framework of a data structure, Facets are not data. "Number of young, female persons" is a Characteristic, which is composed of a measure (number of persons) and two Facets (Young and Female).

As we saw above, labels can be arbitrary tokens that point to descriptions, which are provided in Table I. We call Table I a Dimension Description Data Structure to distinguish it from the Variable Description Data Structure that accompanies Wide format. Table I is linked to Table H by a compound Key consisting of both the Dimension and Facet columns. In Multidimensional format, "Female" has transitioned from data to description. As a row label, it is part of the compound Characteristic "Number of Young, Female persons."

Dimension	Facet	Description
Gender	Male	Identified as “male” by Source
	Female	Identified as “female” by Source
Age	Young	Younger than age 15
	Old	Age 15 or older

Table I. Dimension Description Data Structure for Multidimensional Data

Figure 12 follows the Characteristic “Gender” and the Characteristic Value “Female” from Long to Wide to Multidimensional format in smaller steps to illustrate the semantic transpositions taking place. In Long format (Figure 12.A) “Gender” and “Female” are both Instance Values in the data, and they are clearly identified as a Characteristic and Characteristic Value by their location in the “Property” and “Value” columns respectively. When we move to Wide Format (Figure 12.B), “Gender” is semantically transposed to become a column label, the meaning of which is explained in the Variable Description Data Structure, while “Female” remains a Characteristic Value in the data array.

Figures 12.C and 12.D move from Wide to Multidimensional Format in two steps. The first step (Figure 12.C) converts the Characteristic Gender into two Characteristics Female and Male. Unlike Gender, which has a value domain with text values (“Female”, “Male”), the Characteristic Values of the Female and Male Characteristics are either True or False (shown as 1 and 0), but this does not reduce the information content in the table. When we perform a similar transformation by converting birth and death dates to True/False values for Young and Old, we are losing information by converting exact dates and ages into broader categories. The data structure shown in Figure 12.C will be unfamiliar to most readers, because it is rarely explicit. Most software designed to operate on Wide format data can go from Figure 12.B to Figure 12.D in one step, as we show in the Appendix. We include Figure 12.C, because it shows the semantic transposition of “Female” from a Characteristic Value to a Characteristic without aggregation.

In the next step (Figure 12.D), we aggregate by counting the number of persons in each of the four possible combinations of Gender and Age. At this stage, the identities of individuals are subsumed under a new Characteristic (Count), which is the number of persons with each combination of Gender and Age Group. Since values of Gender and Age Group (Figure 12.D) uniquely identify values of Count, they have become Identifiers that form composite Keys.

In terms of information content, Figure 12.D and Figure 12.E are identical. At this stage the difference between Wide and Multidimensional is in the capabilities of the software in which they are implemented. Software designed for n-cubes and multi-indexes treat Identifiers (e.g., Gender and Age) as Dimensions that facilitate the selection of individual Characteristic Values or subsets of Characteristic Values. In DDI-CDI Gender and Age (Figure 13) are designated Dimension Components to reflect this additional functionality. For example, Figure 12.E can be sliced by Gender to extract a subset of males by age group. Although our example has only two dimensions, Gender and Age Group, we could have added more dimensions, like Reference Area, to produce a 3, 4, or higher dimensional table. Higher dimensional tables have practical applications in data retrieval, but we use only two dimensions to simplify our presentation.

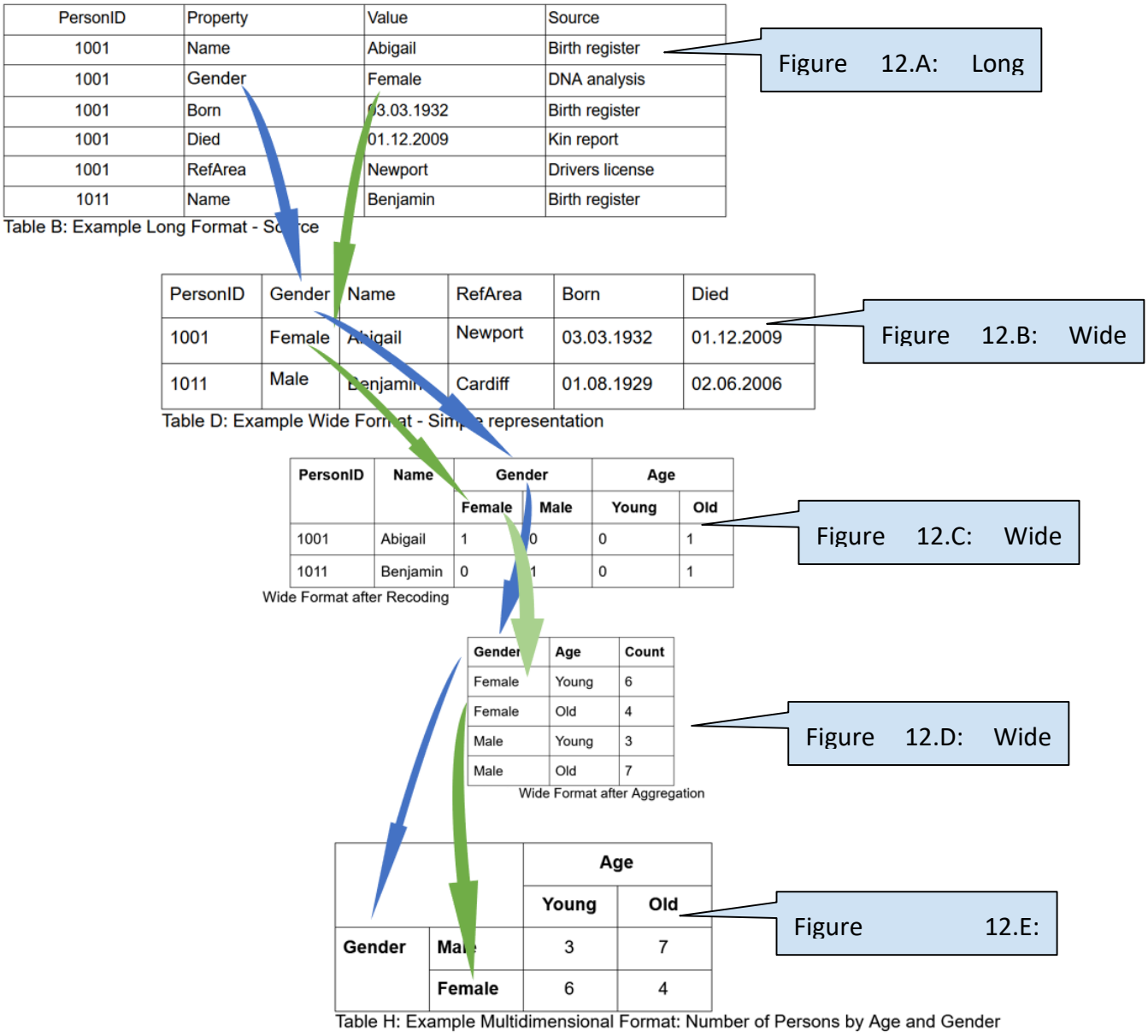


Table B: Example Long Format - Source

Table D: Example Wide Format - Simple representation

Table H: Example Multidimensional Format: Number of Persons by Age and Gender

Figure 12. The Trajectory of Characteristic "Gender" and Characteristic Value "Female" from Long to Wide to Multidimensional Format

The differences between Figure 12.D and Figure 12.E are clearer when we view them from the perspective of the data user. A user viewing Figure 12.D through a spreadsheet or statistical analysis package will see "Female" and "Male" (as well as "Young" and "Old") as Characteristic Values under the "Gender" (or "Age") Characteristic. When software enables the multidimensional aspect of Figure 12.E, the user sees "Female" and "Male" as Facets on the "Gender" Dimension within the descriptive frame of the dataset, where they can be combined into Keys to identify subsets of data, like "Young Females".

The transition from Wide format to Multidimensional format is also depicted in Figure 13 using DDI-CDI components to identify the roles of variables in each format. As we saw in Figure 12, one variable, Gender, moves unchanged to Multidimensional format, and two new variables, Age and Number of Persons, are derived from variables that appear in Wide format. Age is derived from dates of birth and death as described above. Gender and Age, which would have been Measure Components in Wide format are Dimension Components in Multidimensional format. Figure 13 describes Number of Persons as a count of values of Person ID, as one would in an SQL aggregation command. However, counts occur at the intersection of Dimensions as in an SQL Group By clause. (See Appendix for SDTL notation.)

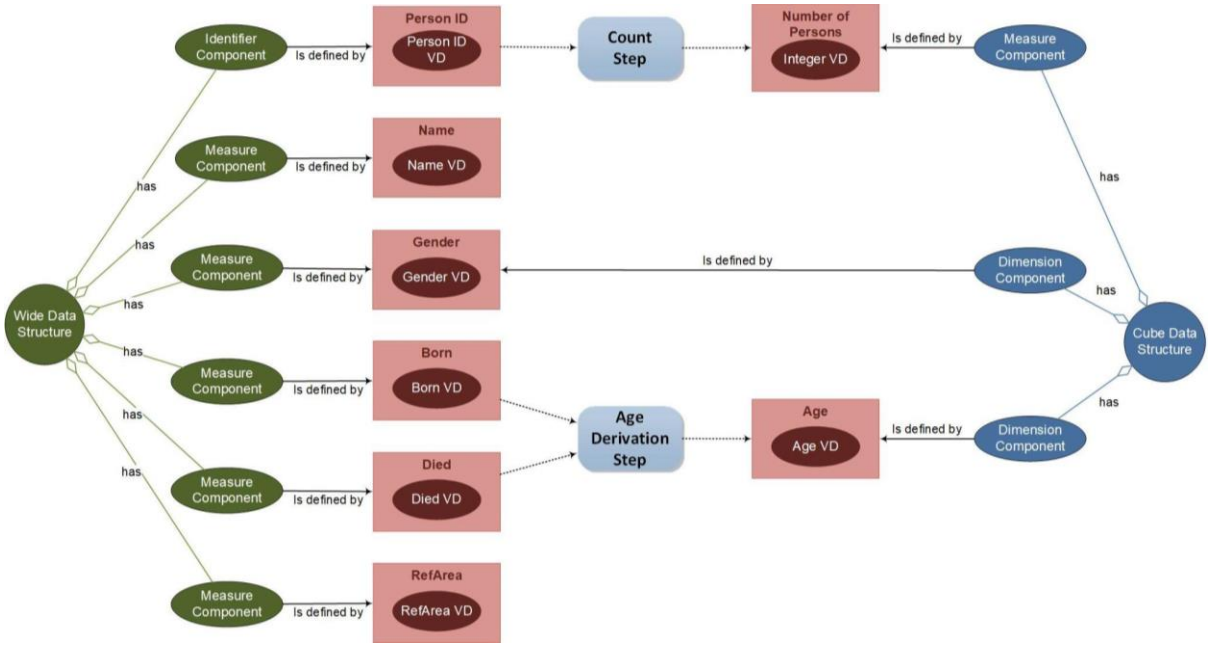


Figure 13 DDI-CDI Representation of Multidimensional Format

Table J provides a subset of the Variable Description Data Structure for the multidimensional data in H. We describe here two Variables, "Count" and "Gender", each of which has three properties (Name, Description, and ValueDomain). We also show that the values within a ValueDomain may have Names and Descriptions. The information represented in Table J is more complex than previous tables, and we present it as "Nested Name-Value Pairs." We use "Nested Name-Value Pairs" to refer to non-rectangular data structures such as XML and JSON. Metadata in standards like DDI, SDMX, and Ecological Markup Language (EML) are often shared in XML or JSON. DDI-CDI does not describe non-

rectangular data arrays like Table J, but we use concepts from DDI-CDI to represent Table J in Figure 14 below. A JSON representation of such a structure is provided in Appendix 3 of this document.

Variable	*Name*	"Count"		
	DescriptiveText	"Number of Persons"		
	ValueDomain	(Set of non-negative integers)		
Variable	*Name*	"Gender"		
	DescriptiveText	"Gender as reported in source document"		
	ValueDomain	Code	*Notation*	"Male"
			DescriptiveText	"Identified as 'Male'"
		Code	*Notation*	"Female"
			DescriptiveText	"Identified as 'Female'"

Table J. Variable Description Data Structure for Multidimensional Format

The simplest name-value pairs consist of a property and a string, such as Name: "Count". However, a group of name-value pairs can be nested inside a value. To understand this data structure, it helps to read Table J from right to left. The first three lines of the table present three simple name-value pairs: Name: "Count", DescriptiveText: "Number of Persons", ValueDomain: (Set of non-negative integers). Taken together, these three pairs are the value for the first Variable, the property named in the left-most column. The Variable named "Gender" is described with three levels of nesting. Reading from right to left and bottom to top, the Notation and DescriptiveText properties are simple name-value pairs, which are nested in a Code. Codes are nested inside a ValueDomain, and ValueDomain is nested inside a Variable. The nesting of Codes inside a ValueDomain makes "Gender" more complex than "Count", but both Variables have the same three properties: Name, DescriptiveText, and ValueDomain.

In Figure 14 we add logical data structures for the descriptive information required to interpret a Multidimensional Data Structure. The panel in the center shows the Cube Data Structure, which is the data array illustrated in Figure 12E and the outcome of the procedures shown in Figure 13. The data consists of three variables. Gender and Age are Dimension Components, and Count is a Measure Component. These variables are linked to variable descriptions through the headers which accompany the data array. In other words, the meaning of the values measured in a data cube must be defined in a different data object, such as a metadata file in SDMX or DDI format.

The panels on the left and right of Figure 14 represent the Variable Description Data Structure found in Table J. Notice that the pink boxes at the top are the Names in the Name-Value schema used in Table J.

The panel on the left of Figure 14 describes Count, the Measure Component in the Cube Data Structure. Recall that Count is the name of the variable created by aggregating over rows in groups defined by values of Gender and Age. Count appears in the header of the data array produced by the aggregation step (Figure 12D), although it is not shown in our illustration of a data cube (Figure 12E). To add descriptive information about the Measure in this Data Cube (center panel), we link Count in the header of the Cube Data Structure to Count as the Name of a Variable in the Variable Description Data structure (left panel). Count has two other properties, Descriptive Text and Value Domain, which are linked to Name: "Count" by descending from the same Variable.

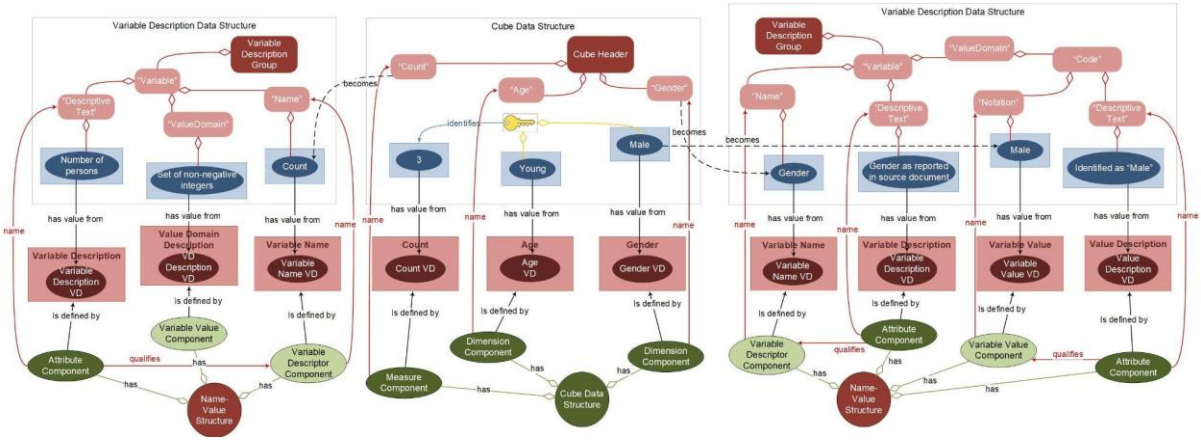


Figure 14. Multidimensional Data Structure with Variable Description Data Structure

The panel on the right illustrates the description of a Facet ("Male") within a Dimension ("Gender"). Note that the row and column headers for Table 11E have two levels, which are both found on the right side of Figure 14. The outer headers ("Gender" and "Age:) are the names of Variables serving as Dimensions. The inner headers ("Male"/ "Female" and "Young"/ "Old") are Facets of the data cube, which are Codes within the ValueDomains of their respective Variables. The nesting of Codes within a ValueDomain, which we showed in Table J is also present in Figure 14. Properties for the Code named "Male" are linked to the ValueDomain of Variable "Gender", and any number of Codes may be part of a ValueDomain.

Converting these data to Multidimensional format is an additional step in the semantic transposition of Datapoints from the data array to the descriptive frame. We showed above that Characteristics (e.g., "Gender" and "Name"), which were data in Long format, become metadata in Wide format. In this section, we showed Characteristic Values (e.g., "Female", "Young") moving from the data array to the descriptive frame as Facets in Multidimensional format. The functions that Facets perform are not identifiable from the data, but from the descriptive frame (metadata) associated with the software. Users recognize that Gender rendered as a Dimension is a different view of the same underlying data as Gender presented as a Measure.

Long Format Revisited

We now apply to Long format two insights from our discussion of Wide format. First, we show that Long format also has an implied Variable Description Data Structure. The column headers in Long format can be arbitrary text that is explained in an accompanying document or dataset. Second, Instance Values in a Long format Data Structure may point to explanations in the Variable Description Data Structure. Moreover, the Variable Description Data Structure may include global resources by using URIs as Instance Values. This means that the Variable Description Data Structure may not be a single physical data file. It could be an array of resources, including published documents, distributed web services, and a history of common practices and traditions.

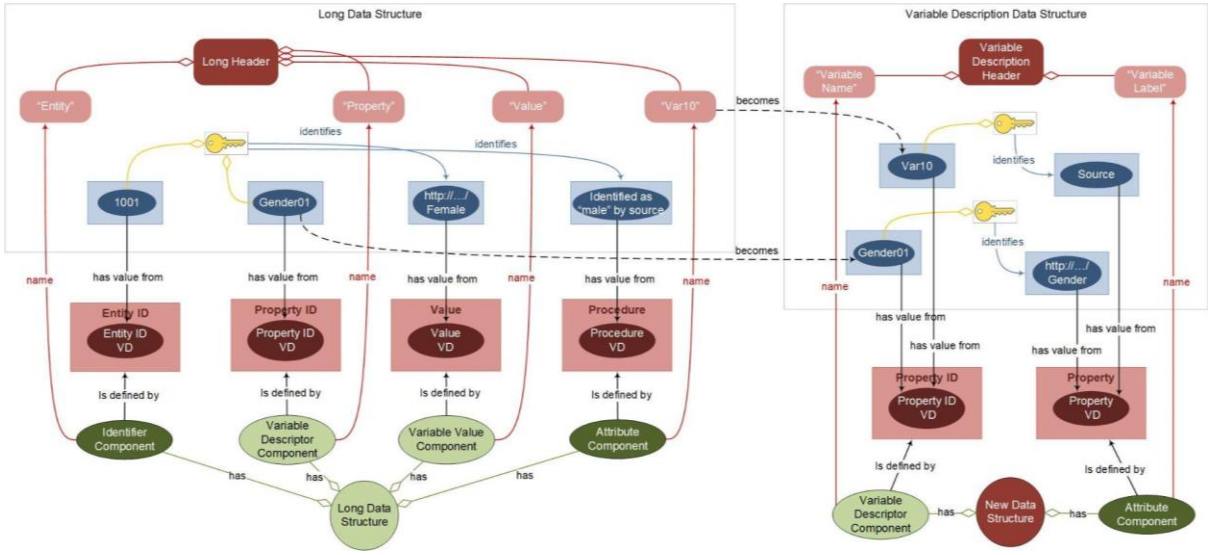


Figure 15. Long Data Structure with Variable Description Data Structure

There are two things to note in Figure 15, which shows a Long Data Structure with a related Variable Description Data Structure similar to the one that we showed in Figure 11 for Wide Format. First, among the pink ovals showing the column headers on the left side of the figure, we have used "Var10" as a column header. "Var10" is then identified as "Source" in the Variable Description Data Structure on the right side of the figure. As we saw with Wide Format in Figure 10, the column headers in a data table may not provide information about the meaning of values in that column. In Figure 12, the meaning of "Var10" is explained in an accompanying Variable Description Data Structure, which is not limited to the technical requirements of a column header. "Var10" could be associated with other attributes, like a definition, citation, instrument model, etc.

We also see that the Characteristic (Variable Descriptor Component) measured in Figure 15 is given as "http://.../Gender" with a value of "http://.../Gender/Female" (see the second and third blue boxes from the left). These URIs are resolved to "Gender" and "Female" in the Variable Description Data Structure on the right side of the diagram. We use this to show not only that a Characteristic in a Long Data Structure can be resolved in the Variable Description Data Structure, but also that the Variable Description Data Structure could be a web service rather than a Data Set. When the variable "Gender" is a link to an online controlled vocabulary, it can resolve not only to a definition of the variable but also to a value domain. In addition, the landing page for the controlled vocabulary can contain a link from the variable to the concept measured by the variable.

Connecting the Dots

As the need to share data across scientific domains increases, differences in languages and practices for constructing and describing data become more important. This paper offers concepts and terminology that can bridge these differences. We bring together two traditions, the Observation and Measurement Model (ISO 19156), which offers a rich relational framework for contextualizing the data creation process, and the elaborate descriptive structures utilized in the DDI and SDMX standards. In particular, we build upon the new DDI-Cross Domain Integration model. We use DDI-CDI to characterize three common data formats Long, Wide, and Multidimensional. However, describing the format of the data is insufficient without simultaneously describing how information about the data is provided, and the O&M and DDI/SDMX traditions diverge radically in that respect. We call the process of changing the representation of both data content and data description "semantic transposition," and we show that the DDI-CDI model can be applied to data description as well as data content. In other words, we show that the boundary between data and metadata is flexible and permeable.

In our terminology, the key difference between Long format data and Wide or Multidimensional format data is the handling of the Characteristic associated with a Characteristic Value. A Characteristic Value is the outcome of an observation, i.e., a number or descriptive text, that is associated with the Characteristic, i.e., a property or attribute. In Long format, which we have used to represent the O&M approach, the Characteristic is included in the data array where the Characteristic Value is found. DDI and SDMX were created to annotate Wide and Multidimensional formats, where the Characteristic is considered "documentation" to be provided separately from the "data." Semantic transposition occurs when re-formatting data implies moving the Characteristic from the data content to the data description frame or vice versa.

We bridge this gap by showing that the Wide and Multidimensional formats imply the existence of a parallel Variable Description Data Structure linking each Characteristic Value to a Characteristic. The Variable Description Data Structure is itself a data array that can be described by the DDI-CDI model. The central contribution of the DDI and SDMX standards is to convert the Variable Description Data Structure from the realm of paper into a machine-actionable object. Thus, we can trace both the Characteristic and the Characteristic Value as the data are transformed from Long to Wide to Multidimensional.

We are not arguing that the DDI-CDI standard needs to develop a separate specification for metadata. Rather, DDI- CDI should be applying the same concepts to data structures containing data and metadata. The terminology provided here is designed to simplify that process by avoiding terms that are used in ambiguous and inconsistent ways, like "attribute". From our point of view, Characteristic Values (data) and Characteristics (metadata) are both Datapoints that can be acted upon by machines as well as people. When we translate data across disciplines, we must recognize that semantic transposition is often reorganizing structures but maintaining the content of the entire dataset -- data and metadata.

Data stewards in the social sciences should be aware that reliance on Wide format data is a disadvantage in a world moving toward FAIR. As we showed, Wide format does not have a way of associating Qualifiers with Characteristic Values. Since there are no inherent relationships among columns in a Wide data structure, nothing links a measure of data quality with the variable that it describes. The only way to make this connection is in the metadata. Even if the relation between these variables is well described in the metadata, parsing XML metadata is not in the skill set of social science researchers. In effect, this requires human intervention and prevents fully automated analysis, which is one of the goals of FAIR.

Recognition of the fluid boundary between data and metadata is essential for achieving the interoperability promised by the FAIR principles. Standards, like O&M, DDI, and SDMX create different ways of encapsulating information and place different boundaries between "data" and "metadata." As we have shown, interoperability often requires semantic transposition, moving information from "data" to "metadata" or the reverse. Thus, interoperability requires mappings between structures that have different understandings of what information belongs in "data" and "metadata." DDI-CDI has created a language for describing these mappings, but it should be extended to support semantic transposition. In practice, the creation of these mappings is further inhibited by different data cultures based on incompatible vocabularies. Our domain-independent vocabulary is intended to enable this much needed cross-domain conversation.

References

Beine, M., Hames, N., Weber, J. H., & Cleve, A. (2014) 'Bidirectional Transformations in Database Evolution: A Case Study At Scale', Paper presented at the EDBT/ICDT Workshops.

Britell, S., Delcambre, L. M., & Atzeni, P. (2016) 'Facilitating data-metadata transformation by domain specialists in a web-based information system using simple correspondences', Paper presented at the International Conference on Conceptual Modeling.

Cox, S. J. D. (2011) 'ISO 19156:2011 Geographic information – Observations and measurements', Retrieved from http://doi.org/10.13140/2.1.1142.3042

DDI Alliance. (2020a) 'DDI-Cross Domain Integration: Detailed Model', Retrieved from https://ddialliance.org/Specification/ddi-cdi

DDI Alliance. (2020b, December 1, 2020) 'Structured Data Transformation Language', Retrieved from https://ddialliance.org/products/sdtl/1.0

Hernández, M. A., Papotti, P., & Tan, W.-C. (2008) 'Data exchange with data-metadata translations', Proc. VLDB Endow., 1(1), 260-273. doi:10.14778/1453856.1453888

ISO 19156:2022 (2022) 'Observations, Measurements and Samples', forthcoming.

ISO/TC 211 Terminology Maintenance Group. (2020) 'ISO/TC 211 Multi-Lingual Glossary of Terms: Entity', Retrieved from https://isotc211.geolexica.org/concepts/1948/

Olivé, Antoni. (2007) 'Conceptual Modeling of Information Systems', Springer Berlin / Heidelberg.

Papotti, P., & Torlone, R. (2009) 'Schema exchange: Generic mappings for transforming data and metadata', *Data & Knowledge Engineering*, 68(7), 665-682. doi:https://doi.org/10.1016/j.datak.2009.02.005

Provenance Working Group. (2013) 'The PROV Namespace'. Retrieved from http://www.w3.org/ns/prov#Entity

San Gil, I., Vanderbilt, K., & Harrington, S. A. (2011) 'Examples of ecological data synthesis driven by rich metadata, and practical guidelines to use the Ecological Metadata Language specification to this end', *International Journal of Metadata, Semantics and Ontologies*, 6(1), 46-55.

Statistical Data and Metadata Exchange (SDMX). (2013, November 11, 2021) 'ISO 17369:2013 Statistical data and metadata exchange (SDMX)'. Retrieved from https://www.iso.org/standard/52500.html

Statistical Data and Metadata Exchange (SDMX). (2021) 'SDMX'. Retrieved from https://sdmx.org/

Vardigan, M., Heus, P., & Thomas, W. (2008) 'Data documentation initiative: Toward a standard for the social sciences', *International Journal of Digital Curation, 3*(1).

Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.-W., da Silva Santos, L. B., Bourne, P. E. (2016) 'The FAIR Guiding Principles for scientific data management and stewardship', *Scientific Data*, 3.

Wyss, C. M., & Robertson, E. L. (2005) 'A formal characterization of PIVOT/UNPIVOT', Paper presented at the *Proceedings of the 14th ACM international conference on Information and knowledge management, Bremen, Germany*. https://doi.org/10.1145/1099554.1099709

Xue, J., Shen, D., Nie, T., Kou, Y., & Yu, G. (2013, 10-15 Nov. 2013) 'Inferring and Propagating Pivot Dependencies in Schema Transformation between Data and Metadata', Paper presented at the 2013 10th Web Information System and Application Conference.

Appendix 1: Semantic Transposition

Semantic Transposition: (aka Flip-Flop) in refactoring from a conceptual model, the lateral transposition of the representation of a characteristic from the data structure to the data content, that retains isomorphic coherence between representations

Semantic Transposition is the concept, Semantic Transposition Refactoring (STR) the process

- Simple:

```
{
    "@context":                                          {
      "wind-speed":                    "http://...wind-speed",
      "FoI":              "http:            //...OGC/O&M/FoI",
      "geom":                             "http://...OGC/geom"
    },
    "FoI":                                                1,
    "geom":                                           "XXX",
    "wind-speed":                                         5
}
```

- Complex:

```
{
    "@context":                                          {
      "FoI":              "http:            //ogc.../O&M/FoI",
      "observedProperty":             "http://...OGC/obsProp",
      "result":                       "http://...OGC/result",
      "geom":                             "http://...OGC/geom"
    },
    "FoI":                                                1,
    "geom":                                           "XXX",
    "observedProperty":             "http://...wind-speed",
    "result":                                             5
}
```

 Alter, George; Rizzolo,Flavio; and Schleidt, Kathi (2023) View points on data points, IASSIST Quarterly 47(1), pp. 1-39. DOI: https://doi.org/10.29173/iq1051

Appendix 2: Data Transformations and Aggregations

As described above in the section on multidimensional formats, to obtain counts within categories, such as age groups, properties that provide a continuous value range must be transformed or recoded to a related property represented by a discrete set of values. Once all relevant properties have been transformed to ones suitable for grouping, the aggregation can be performed.

We use a simplified version of Structured Data Transformation Language (SDTL; DDI Alliance, 2020b) to describe data transformations. SDTL provides machine-actionable descriptions (i.e., provenance metadata) of variable-level processes like this. (See https://ddialliance.org/products/sdtl/1.0)

Transformation

In some cases, the properties being provided for individuals circumscribe the actual property of interest. In our primary demographic data, we have dates of birth and death, but we are actually interested in the age attained by the individual at the time of death. Age at death is implicit in the data, and we must perform a calculation on the values provided for birth and death dates to obtain a new property for each individual.

SDTL for computing age at death from birth and death dates:

```
Command: Compute
    Variable: Age
    Expression:
        Function: Division
            ArgumentName:  EXP1
                ArgumentValue:
                    Function: Subtraction
                        ArgumentName:  EXP1
                            ArgumentValue: Died
                        ArgumentName:  EXP2
                            ArgumentValue: Born
            ArgumentName: EXP2
                ArgumentValue:
                        TimeDurationConstant:
                                TimeDurationValue: “P365.25D”
```

Classification, Recoding and Transformation

Under “recoding” we understand the process of assigning discrete values, usually from a classification system, to serve as a representative of a property that has been provided by a continuous value range.

A simple example of this concept pertains to the age of an individual calculated above from the dates of birth and death. In order to provide a discrete age axis within our multidimensional representation of the data, this property must be transformed to a related property with a discrete value range. In our example, the final age_group property is defined with the following values:

- Child: Age <= 14 Years

- Adult: 15 <= Age <= 64
- Old: Age >= 65

Recoding is simply a matter of determining which of the defined groups the value provided for the individual belongs to, this value is then assigned to the individual via the new Age_group property.

SDTL for recoding into age groups:

```
Command: Recode
    Recoded variables:
        Source: Age
        Target:  Age_group
    Rules:
        RecodeRule:
            FromValue: 0
            To: 14
            Label: Child
        RecodeRule:
            FromValue: 15
            To: 64
            Label: Adult
        RecodeRule:
            FromValue: 65
            To: NumericMaximumValueExpression
            Label: Old
```

Aggregation

SDTL description of the aggregation process:

```
Command: Collapse
    GroupByVariables: Gender, Age_group
    AggregateVariables:  Compute count = col_count(Name)
    ProducesDataframe:
        DataframeDescription:
            DataframeName: cube20
            RowDimensions: Gender, Age_group
```

 Alter, George; Rizzolo,Flavio; and Schleidt, Kathi (2023) View points on data points, IASSIST Quarterly 47(1), pp. 1-39. DOI: https://doi.org/10.29173/iq1051

Appendix 3: JSON Representation

Variable Description Data Structure for Multidimensional Format

The Variable Description Data Structure for Multidimensional Format could be represented in JSON as shown below.

```
{
    "Variable": {
            "Name": "Count",
            "DescriptiveText": "Number of Persons",
            "ValueDomain": ["(Set of non-negative integers)"]
    }
}, {
    "Variable": {
            "Name": "Gender",
            "DescriptiveText":   "Gender   as   reported   in   source
document",
            "ValueDomain": [{
                    "Code": {
                            "Notation": "Male",
                            "DescriptiveText":   "Identified   as
'Male'"
                    }
            }, {
                    "Code": {
                            "Notation": "Female",
                            "DescriptiveText":   "Identified   as
'Female'"
                    }
            }]
    }
}
```

Endnotes

[1] George Alter is Research Professor Emeritus in the Institute for Social Research at the University of Michigan. He can be reached by email: altergc@umich.edu.

[2] Flavio Rizzolo is Senior Data Science Architect for Statistics Canada. He can be reached by email: flavio.rizzolo@statcan.gc.ca.

[3] Kathi Schleidt is a data scientist with a specialty in environmental informatics. She is the founder of DataCove e.U. She can be reached by email: kathi@datacove.eu.

[4] Ideally, this should be a URI providing further information such as possible values the methodology can provide.