PharmaSUG 2014 - Paper DS13

How Valued is Value Level Metadata?

Shelley Dunn, d-Wise, Morrisville, NC

ABSTRACT

One of the challenges of implementing SDTM and ADaM is the vertical data structure where some variables are dependent on test code (xxTESTCD) in SDTM or Parameter (PARAMCD) in ADaM. This type of metadata is described as "value level metadata" and at a basic level describes the metadata of one variable based on the value of another variable. For example, how do you describe the result of a Vital Sign (VSORRES) when the Vital Signs test code (VSTESTCD) is WEIGHT? This definition can be different for different values of the test code and is best described based on those values.

The underlying question is whether this type of value level metadata is important and what value it provides to stakeholders. Stakeholders include a wide range of people such as study specification creators, SAS programmers, standards developers, and FDA reviewers, each of which might have different needs. Defining value level metadata can go from one end of the spectrum where you define it for every variable to the other end where you define none at all. What criteria provide the best practice for determining when to use value level metadata? The CDISC Define-XML Specification Version 2.0 document provides some general, albeit vague, rules.

- "Value Level Metadata **should be** provided when there is a need to describe differing metadata attributes for subsets of cells within a column."
- "It is **not required** for Findings domains where the results have the same characteristics in all records, such as IE domains."
- "In ADaM, value level metadata **often describes** AVAL or AVALC in BDS data structures based on values of PARAMCD."
- "Value Level Metadata **should be** applied when it provides information useful for interpreting study data. It need not be applied in all cases."
- "It is left to the **discretion of the creator** when it is useful to provide Value Level Metadata and when it is not."

The overriding message is that there are few requirements for what variables require this metadata and most of the criteria are based on the subjective notion of providing useful information. With requirements open to interpretation there are many correct ways to apply this metadata. What is considered useful to one stakeholder may or may not be useful to another.

This presentation will use experience from a range of projects to look at why, when, and how to define value level metadata balancing the amount of effort it takes to define this information with its value to stakeholders.

INTRODUCTION

With all the information about standards it is surprising how few standards or rules there are governing value level metadata (VLM). There are many layers and challenges associated with the concept of VLM. First of all, what is it?

At the very basic level VLM is a relationship between two variables. This brings up additional questions. Such as, how does one know when variables have this relationship? How is this relationship between variables defined? Why do we need to define this relationship? How is the metadata describing this relationship used? And who/how does this help stakeholders better understand and interpret data in a meaningful way?

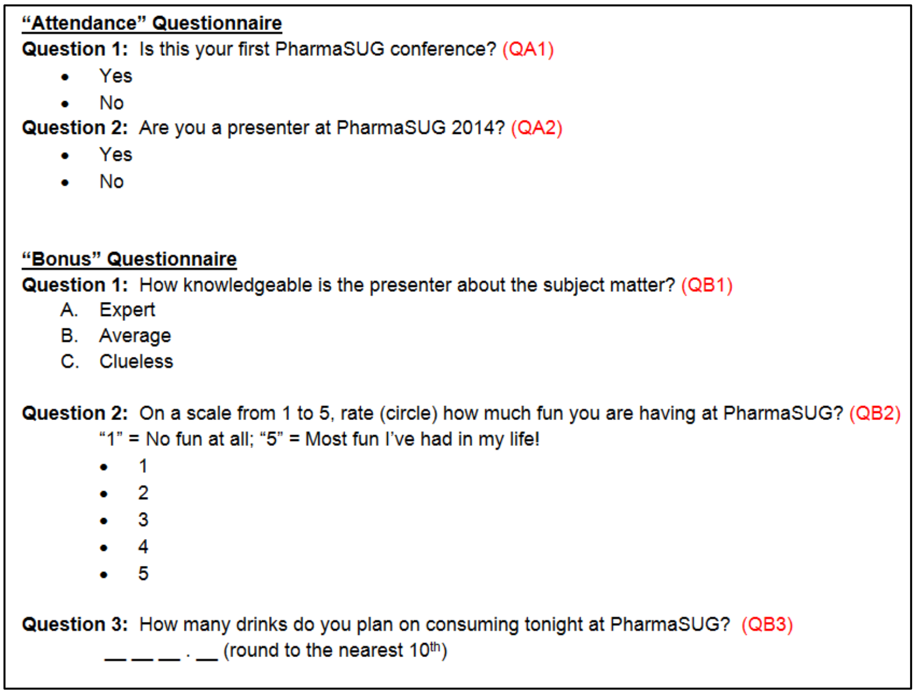
This paper will explore each of these questions and provide examples to clearly illustrate when VLM is applicable and appropriate. There are different types of relationships between variables and one objective of this paper is to categorize these. These categories can then provide language to use to determine the value of VLM. A global look at how metadata is stored and generated will provide an additional layer of complexity.

HORIZONTAL AND VERTICAL STRUCTURES

To really understand what is meant by VLM it is necessary to first understand the difference between a horizontal data structure and a vertical data structure. Simply stated, horizontal means the data structure tends to be wide and has many variables; vertical means the data structure is tall and skinny and has fewer variables. Within a horizontal structure there is typically a different variable for every data point that is collected and the name of the variable can aid in defining the content. In a vertical structure similar types of variables are grouped into a single variable and descriptive variables are added to define the contents.

The following Questionnaire Case Report Form (CRF) example, see *Figure 1*, will be used throughout this paper to illustrate both horizontal and vertical data structures as well as the metadata associated with each. Refer to the section “EASY FIGURE AND TABLE REFERENCES” at the end of this paper for a short description of each.

Figure 1



“Attendance” Questionnaire

Question 1: Is this your first PharmaSUG conference? (QA1)

- Yes
- No

Question 2: Are you a presenter at PharmaSUG 2014? (QA2)

- Yes
- No

“Bonus” Questionnaire

Question 1: How knowledgeable is the presenter about the subject matter? (QB1)

A. Expert
B. Average
C. Clueless

Question 2: On a scale from 1 to 5, rate (circle) how much fun you are having at PharmaSUG? (QB2)

“1” = No fun at all; “5” = Most fun I’ve had in my life!

- 1
- 2
- 3
- 4
- 5

Question 3: How many drinks do you plan on consuming tonight at PharmaSUG? (QB3)

___ ___ ___ . ___ (round to the nearest 10th)

Figure 1. Questionnaire Case Report Form (CRF)

There are two different questionnaires on the CRF. Questionnaire A has two questions (QA1 & QA2) about PharmaSUG “Attendance” and Questionnaire B has three (QB1, QB2, AB3) PharmaSUG “Bonus” questions.

Table 1

STUDY	SUBJECT	QA1	QA2	QB1	QB2	QB3
ABC-123	101	Yes	No	A	3	2
ABC-123	102	No	No	C	1	6
ABC-123	103	No	Yes	A	5	3.2

Table 1. Representing CRF data horizontally: DATASET = QUESTION

Table 1 contains data for three subjects representing in a horizontal data structure. The name of the data set is QUESTION. The variable QA1 refers to Questionnaire A question #1 and QB3 refers be Questionnaire B question #3, and so on. The variable names are used to describe the contents. The value of any single variables in the above structure has no bearing on the value of any other variable within the data set. Data collection systems and their associated databases have been designed to collect horizontal data because they collect single points in time for a single subject and not aggregated data.

Table 2

STUDYID	USUBJID	QSTESTCD	QSTEST	QSORRES
ABC-123	101	QA1	Questionnaire A1: Is this your first PharmaSUG conference?	Y
ABC-123	101	QA2	Questionnaire A2: Are you a presenter at PharmaSUG 2014	N
ABC-123	101	QB1	Questionnaire B1: How knowledgeable is the presenter about the subject matter?	A

Table 2 continued

STUDYID	USUBJID	QSTESTCD	QSTEST	QSORRES
ABC-123	101	QB2	Questionnaire B2: On a scale from 1 to 5, rate (circle) how much fun you are having at PharmaSUG?	3
ABC-123	101	QB3	Questionnaire B3: How many drinks do you plan on consuming tonight at PharmaSUG?	2
ABC-123	102	QA1	Questionnaire A1: Is this your first PharmaSUG conference?	N
ABC-123	102	QA2	Questionnaire A2: Are you a presenter at PharmaSUG 2014	N
ABC-123	102	QB1	Questionnaire B1: How knowledgeable is the presenter about the subject matter?	C
ABC-123	102	QB2	Questionnaire B2: On a scale from 1 to 5, rate (circle) how much fun you are having at PharmaSUG?	1
ABC-123	102	QB3	Questionnaire B3: How many drinks do you plan on consuming tonight at PharmaSUG?	6
ABC-123	103	QA1	Questionnaire A1: Is this your first PharmaSUG conference?	N
ABC-123	103	QA2	Questionnaire A2: Are you a presenter at PharmaSUG 2014	Y
ABC-123	103	QB1	Questionnaire B1: How knowledgeable is the presenter about the subject matter?	A
ABC-123	103	QB2	Questionnaire B2: On a scale from 1 to 5, rate (circle) how much fun you are having at PharmaSUG?	5
ABC-123	103	QB3	Questionnaire B3: How many drinks do you plan on consuming tonight at PharmaSUG?	3.2

Table 2. Table Representing CRF data vertically: DATASET = QS

Table 2 contains the exact same data as *Table 1* but in a vertical data structure. The name of this data set is QS to differentiate it from the previous example. The variable QSTESTCD is use to combine all of the variables QA1, QA2, QB1, QB2, and QB3 shown in *Table 1*. Compare the grey highlighted cell in *Table 1* to the grey highlighted cells in *Table 2* to see how the horizontal data set QUESTION and the vertical data set QS are related. In *Table 2* when the variable QSTESTCD = "QA1" the value of QSORRES is equal to the value of *Table 1* QA1.

Within *Table 2*, compare the results of QSTESTCD with the results of QSORRES. Specifically when QSTESTCD = "QA1" the value of QSORRES can only take on the values of "Y" or "N," and when QSTESTCD = "QB1" the value of QSORRES can only take on the values of "A," "B," or "C." Due to the structure of the vertical data set there is a relationship between the values of QSTESTCD and the values of QSORRES.

Both horizontal and vertical data provide different benefits and drawbacks. One benefits of a horizontal structure is that the programmer or developer is not confined to rules governing naming conventions and structure. Another benefit is that there is generally no need to define one variable in terms of another. While it is possible to make a horizontal structure into a data standard, one of the downsides is that this particular data set is inconsistent with current industry standards. Other potential downsides of a horizontal structure are the different variable names used across data sets that capture result findings such as questionnaires, vital signs, lab results, and ECGs. Data sets with different variable naming conventions and a different number of variables per data set can add to the complexity of creating re-usable code.

One of the main benefits of a vertical structure, such as those used to capture results data, is that this structure is in line with the current industry standards. Utilizing the same structure and naming conventions across multiple data sets and studies lends itself more easily to using re-usable code for creating data sets and generating analysis output. One of the drawbacks of a vertical structure is that it may not be consistent with the way data is collected. A vertical data set requires additional metadata to describe variables with relationships where variables can contain values that differ in some attribute. For example, instead of collecting each question of a questionnaire as a separate variable, all of the questionnaire answers will be stored within the same variable.

WHAT IS VALUE LEVEL METADATA (VLM)?

Metadata, or data about data, can be broken into four main types: data set or table level, variable level, value level (VL), and controlled terminology (CT). While CT is its own bucket, to better visualize where it comes from the examples below split out CT for variable level and CT for value level.

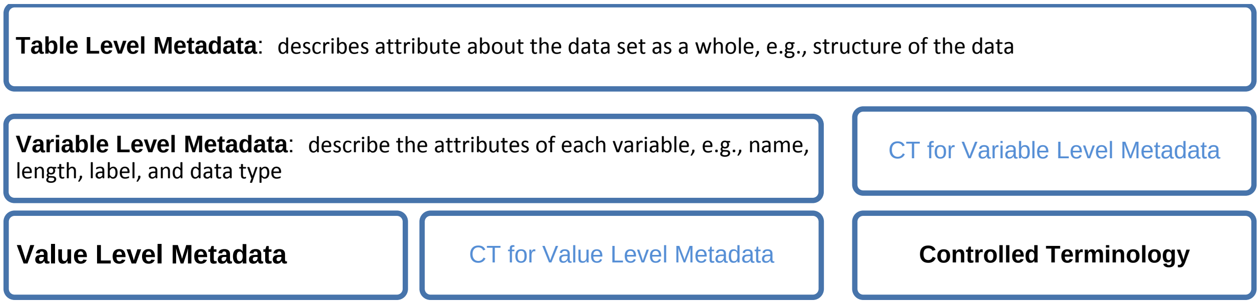


Table Level Metadata: describes attribute about the data set as a whole, e.g., structure of the data
Variable Level Metadata: describe the attributes of each variable, e.g., name, length, label, and data type
CT for Variable Level Metadata
Value Level Metadata
CT for Value Level Metadata
Controlled Terminology

Value Level Metadata (VLM): similar to variable level metadata, it describes attribute such as name, length, label, and data type. The difference between variable level and value level is that variable level provides information about a single variable and value level provides information about the value of single variable in relationship to the value of another variable(s). VLM in a vertical structure defines what would otherwise be a variable in a horizontal structure.

Controlled Terminology (CT): is a list of all the possible values of a specific "variable" (i.e., CT for Variable Level) or all of the possible values of a specific "value of a variable" (i.e., CT for Value Level)

CT for Variable Level
+
CT for Value Level
=
CT

There is some overlap between variable level metadata and VLM. Additionally both VLM and CT describe the values of a variable. Understanding these similarities and differences amongst these types of metadata will aid greatly in determining when to use variable level metadata vs. VLM and/or when to use VLM vs. CT.

A closer look at the metadata associated with a horizontal data structure is shown below. **Note:** examples are not meant to include all possible metadata sponsors may choose to define.

Table 3

DATA SET	STRUCTURE
QUESTION	One record per subject

Table 3. Table Level Metadata for horizontal data set QUESTION

Table 4

VARIABLE	DATA TYPE	LENGTH	LABEL	CT	Format
DATA SET	text	8	Name of Data Set		
STUDY	text	7	Name of Study		
SUBJECT	integer	8	Subject Number		
QA1	text	3	Question A1	Yes, No	
QA2	text	3	Question A2	Yes, No	
QB1	text	1	Question B1	A = Expert, B = Average, C = Clueless	
QB2	integer	8	Question B2	1, 2, 3, 4, 5	
QB3	float	8	Question B3		5.1

Table 4. Variable Level Metadata for horizontal data set QUESTION

In a horizontal structure, such as *Table 4*, only table level and variable level metadata, and CT are necessary to describe the data set. That is, there is no need to provide VLM.

Table 5

DATA SET	STRUCTURE
QS	One record per questionnaire question per subject

Table 5. Table Level Metadata for vertical data set QS

Compare *Table 3* to *Table 5*. In *Table 3* all data for a subject is on a single row; whereas, in *Table 5* the data for each subject is on more than one row and each row is defined by both the subject and the QS question.

Table 6

VARIABLE	DATA TYPE	LENGTH	LABEL	CODELIST	FORMAT
STUDYID	Text	7	Study Identifier		
DOMAIN	Text	2	Domain Abbreviation		
USUBJID	Text	3	Unique Subject Identifier		
QSTESTCD	Text	8	Question Short Name	CL.QSTESTCD	
QSTEST	Text	100	Question Name	CL.QSTEST	
QSORRES	Text	5	Finding in Original Unit	(see below)	(see below)

Table 6. Variable Level Metadata for vertical data set QS

Table 4 and *Table 6* show the variable level metadata for the horizontal and vertical data examples respectively. The CT associated with the horizontal variable level metadata in *Table 4* is contained in the metadata column CT. The CT associated with the vertical variable level metadata in *Table 6* is shown in *Tables 7a, 7b, and 7c.*

Table 7a

CODELIST	SUBMISSION VALUE	DECODE
CL.QSTESTCD	QA1	Question A1: Is this your first PharmaSUG conference?
CL.QSTESTCD	QA2	Questionnaire A2: Are you a presenter at PharmaSUG 2014
CL.QSTESTCD	QB1	Questionnaire B1: How knowledgeable is the presenter about the subject matter?
CL.QSTESTCD	QB2	Questionnaire B2: On a scale from 1 to 5, rate (circle) how much fun you are having at PharmaSUG?
CL.QSTESTCD	QB3	Questionnaire B3: How many drinks do you plan on consuming tonight at PharmaSUG?

Table 7a. CT for variable level metadata for vertical data set QS – QSTESTCD

Table 7b

CODELIST	SUBMISSION VALUE
CL.QSTEST	Questionnaire A1: Is this your first PharmaSUG conference?
CL.QSTEST	Questionnaire A2: Are you a presenter at PharmaSUG 2014
CL.QSTEST	Questionnaire B1: How knowledgeable is the presenter about the subject matter?
CL.QSTEST	Questionnaire B2: On a scale from 1 to 5, rate (circle) how much fun you are having at PharmaSUG?
CL.QSTEST	Questionnaire B3: How many drinks do you plan on consuming tonight at PharmaSUG?

Table 7b. CT for variable level metadata for vertical data set QS – QSTEST

Table 7a is the codelist CL.QSTESTCD for the variable QSTESTCD. The decoded value of the QSTESTCD submission values are the QSTEST submission values. There is a one-to one-relationship between QSTESTCD and

QSTEST. *Table 7b* shows the codelist QL.QSTEST for the variable QSTEST. Consider if this codelist provides any additional information not included in CL.QSTESTCD.

No codelist is assigned to the variable QSORRES; however, if it might look the codelist CL.QSORRES in *Table 7c* below.

Table 7c (Not a best practice – but included to make a point)

CODELIST	**SUBMISSION VALUE**	**DECODE**
CL.QSORRES	Y	Yes
CL.QSORRES	N	No
CL.QSORRES	A	Expert
CL.QSORRES	B	Average
CL.QSORRES	C	Clueless
CL.QSORRES	1	
CL.QSORRES	2	
CL.QSORRES	3	
CL.QSORRES	4	
CL.QSORRES	5	

Table 7c. CT for Variable Level (compare to VLM)

There are several issues associated with defining the codelist CL.QSORRES:

1. The variable level metadata in Table 6 above doesn't explain the difference in the metadata associated with the values of QSORRES when QSTESTCD takes on different values

2. Creating a single codelist for QSORRES doesn't account for specific codelists, e.g., CL.YN, for different values of QSTESTCD

3. Compliance checking using only CT to describe QSORRES and not VLM could cause inaccurate results. For example, the possible results for Question A1 are "Y" and "N" and the possible results for Question B2 are "A," "B," "C." Grouping these codelists together may not allow compliance checks to capture a case where Question A1 has the value "A." While "A" is in the codelist for QSORRES it is only applicable when QSTESTCD = "QA1"

Table 8a

VARIABLE	**WHERE**	**DATA TYPE**	**LENGTH**	**CODELIST** (see *Table 8b* for decodes)	**FORMAT**
QSORRES	QSTESTCD = "QA1"	text	1	CL.YN	
QSORRES	QSTESTCD = "QA2"	text	1	CL.YN	
QSORRES	QSTESTCD = "QB1"	text	1	CL.QB1	
QSORRES	QSTESTCD = "QB2"	integer	1	CL.QB2	
QSORRES	QSTESTCD = "QB3"	float	5		5.1

Table 8a. VLM for vertical data set QS

Table 8a above illustrates a best practice for producing VLM for the variable QSORRES based on the values of QSTESTCD. Metadata associated with values of QSORRES based on QSTESTCD contain different metadata. Specifically the following metadata can and does vary for data type, length, codelist, and format.

Table 8b

CODELIST	SUBMISSION VALUE	DECODE
CL.YN	Y	Yes
CL.YN	N	No
CL.QB1	A	Expert
CL.QB1	B	Average
CL.QB1	C	Clueless
CL.QB2	1	
CL.QB2	2	
CL.QB2	3	
CL.QB2	4	
CL.QB2	5	

Table 8b. CT for vertical data set QS

Table 8b contains the codelist metadata associated with the codelist names identified in the VLM in *Table 8a*. Note that there is not a single codelist for QSORRES but rather each value of QSORRES where QSTESTCD takes on a given codelist is described.

Table 8c

VARIABLE	WHERE	DATA TYPE	LENGTH	VALUES	FORMAT
QSTEST	QSTESTCD = “QA1”	text	100	Question A1: Is this your first PharmaSUG conference?	
QSTEST	QSTESTCD = “QA2”	text	100	Questionnaire A2: Are you a presenter at PharmaSUG 2014	
QSTEST	QSTESTCD = “QB1”	text	100	Questionnaire B1: How knowledgeable is the presenter about the subject matter?	
QSTEST	QSTESTCD = “QB2”	text	100	Questionnaire B2: On a scale from 1 to 5, rate (circle) how much fun you are having at PharmaSUG?	
QSTEST	QSTESTCD = “QB3”	text	100	Questionnaire B3: How many drinks do you plan on consuming tonight at PharmaSUG?	

Table 8c. VLM for vertical data set QS – QSTEST

Table 8c above illustrates how to create VLM for the variable QSTEST based on the values of QSTESTCD. Comparing this VLM in *Table 8c* to the variable level metadata in *Table 6* there is no difference with regards to the metadata fields for data type, length, or format.

Either a horizontal data structure or a vertical data structure can be used to capture the same data. What varies between these two types of structures is tied mainly to the need for VLM to describe the relationships created by the vertical structure that do not exist in the horizontal structure. Determining what metadata is relevant in a horizontal structure is far easier than determining what metadata is relevant in a vertical structure due to the absence of VLM when data is presented horizontally.

WHAT IS THE POINT OF DEFINING VLM?

There is a “Temptation” to title this section “VLM – WHAT IS IT GOOD FOR?”; but, I think we all know the answer to that! At times “absolutely nothing!’” might be the best response.

There are a few main reasons one might want to understand the complexities associated with defining and using VLM.

1. To generate a Define.xml with the idea of providing valuable information to a reviewer
2. To provide instructions to a programmer on how to create certain variables
3. To pre-define metadata standards or create metadata libraries to enable users to create a Define.xml downstream
4. Promote consistency across studies for integrated analysis

DEFINE.XML

The language in the Define.xml documentation indicates that VLM “should,” at the discretion of the creator, be applied to results data where test codes have different attributes as shown in the vertical data structure above. The bottom line is that there are few actual requirements associated with a define.xml when it comes to which variables must include VLM within a Define.xml.

It is important to consider what metadata a reviewer would find useful. For example, in an ADaM BDS data set where “last observation carried forward” is used, it is not enough to set DTYPE = “LOCF” and apply a label. In this example there needs to be a description of what constitutes the “last observation” and in what instances it is “carried forward.” In this case the clarity of an analysis algorithm will provide value to both a programmer to ensure it is programmed correctly as well as a reviewer to ensure there is a clear understanding of the analysis. Deciding what metadata to include is not just about the requirements; but rather, metadata is meant to provide to provide useful information to the end user.

PROGRAMMER INSTRUCTIONS

Algorithms describing programming instructions can also vary for certain variables. If a date variable is generated from a variety of sources and there are algorithms to describe each derivation, a sponsor may elect to define a date variable in terms of different test codes. Would it then be helpful or useful to provide programming instructions in a Define.xml to submit to the FDA? Is it necessary?

PRE-DEFINED METADATA STANDARDS AND CT/VL LIBRARIES

This is of particular interest to companies where data standards can be created and applied across therapeutic areas, compounds, and studies. Other companies may not have the luxury of using a metadata repository, defining standards up front, and establishing CT/VL libraries. For example, some CROs are faced with creating non-standard data sets based on each sponsor’s unique specifications. As a result the generation of metadata for a Define.xml may not be defined until after database lock. Other companies may create all of the metadata prior to opening a study by using pre-defined standards. As part of the exercise to prepare study specifications a global library or metadata repository (MDR) may allow standard references to auto-generate CT and VLM. Although this involves a great deal of work on the front end to establish, the result downstream is that all of the metadata will have been defined prior to database lock. Then it becomes the proverbial “push button” to generate a Define.xml for submission. A later example will show how defining these relationships up front can save time on the backend.

Although most companies tackle VLM as a means for creating a Define.xml or providing additional information to programmers the song remains the same. Just try replacing “War” with “VLM.”

War, huh, yeah
What is it good for
Absolutely nothing
Uh-huh
War, huh, yeah
What is it good for
Absolutely nothing
Say it again, y'all*****

*****Edwin Starr lyrics

DEFINING TYPES OF RELATIONSHIPS BETWEEN VARIABLES

Just because a variable can be described in terms of another variable does not always imply that it should be. It is not enough to know what VLM is; it is important to recognize differences within these relationships. Consider the following terms coined for this paper:

- **Micro-Metadata:** Metadata defined on a variable by variable basis
- **Macro-Metadata:** Sum total of metadata within a system or a set of standards

MICRO-METADATA

No Relationship

The value of one variable does not determine the value of another variable. For example, in *Table 1* above there is no relationship amongst the variables QA1, QA2, QB1, QB2, and QB3. That is, by knowing the value of QA1, “Is this your first PharmaSUG conference?” is “Yes” this in no way determines the values of any other variable.

1:1 Relationship

In *Table 2* the relationship between the variables QSTESTCD and QSTEST is a 1:1 relationship. That is, every value of QSTESTCD maps to exactly one value of QSTEST. There is no difference between the variable level attributes and the value level attributes for QSTEST. Additionally (see *Table 7a* and *Table 8c*) the same relationship information can be gained by looking at either the CT for QSTESTCD or the VLM for QSTEST. It is redundant to include both.

Subset of Table 7a and 8c

CODELIST	SUBMISSION VALUE	DECODE	VARIABLE	WHERE	VALUES
CL.QSTESTCD	QA1	Question A1: Is this your first PharmaSUG conference?	QSTEST	QSTESTCD = “QA1”	Question A1: Is this your first PharmaSUG conference?

Subset of Table 7a – CT for QSTESTCD **Subset of Table 8c – VLM for QSTEST**

Which to include in this case, CT or VL, depends on several factors. This is a great place for standards governance to step in. One might consider including all 1:1 relationships, not including any 1:1 relationships, or some combination of 1:1 relationships.

1:1

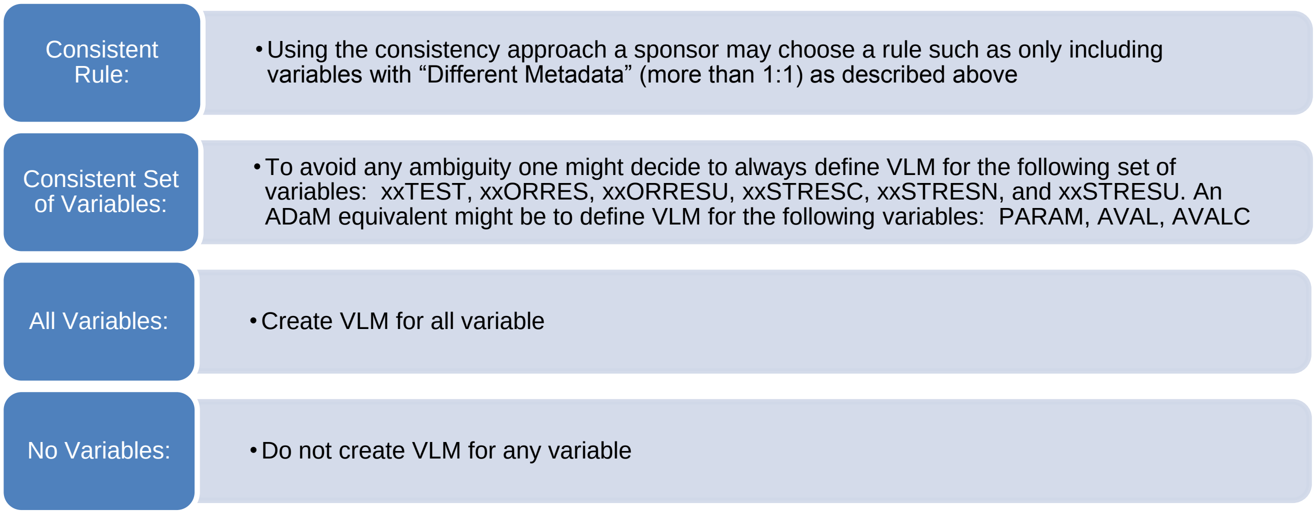
Different Metadata Relationship

In contrast to the 1:1 relationship where variable and value level are the same, “different metadata” is the relationship between two or more variables where more than just the codelist is needed to define all of the metadata. *Table 8a* illustrates how the variable QSORRES has different data type, length, codelist, and format differ when QSTESTCD takes on different values. The different metadata relationship is the primary reason for describing metadata at this level.

MICRO-METADATA

Consistency Approach

The previous examples considered variables on an individual basis. Consistency focuses on a more global approach to defining value level attributes. Consider the following for defining consistency:



Consistent Rule:

- Using the consistency approach a sponsor may choose a rule such as only including variables with “Different Metadata” (more than 1:1) as described above

Consistent Set of Variables:

- To avoid any ambiguity one might decide to always define VLM for the following set of variables: xxTEST, xxORRES, xxORRESU, xxSTRESC, xxSTRESN, and xxSTRESU. An ADaM equivalent might be to define VLM for the following variables: PARAM, AVAL, AVALC

All Variables:

- Create VLM for all variable

No Variables:

- Do not create VLM for any variable

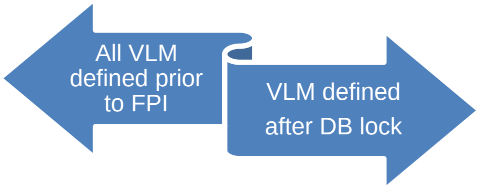
The distinction among these options brings up an important point about VLM. In the “Consistent Rule” option above the variable QSTEST (refer to *Table 6*) would not be described at the value level because the relationship between QSTEST and QSTESTCD is a 1:1 relationship. In the “Consistent Set of Variables” option above the variable QSTEST would be describe at the value level because the relationship between QSTEST is one of the predetermined variables.

While probably not a best practice, either of the last two options are viable options. There is no documentation indication that you cannot describe every variable in relation to another variable. When two variables either do not have a relationship or the relationship is 1:1 the variable level metadata and the VLM, other that the submission values of the variables, will be identical. The final option of never providing VLM can leave holes in providing useful information for a reviewer (see *Table 7c*). Neither are very practical nor elegant solutions but they are possibilities.

The point is that the industry guidance does not clearly define when, if ever, one needs to describe the metadata for variables at the value level. Define.xml documentation does not delineate the different relationships nor does it designate requirements for specific variables or types of variables that need to use VLM. The decisions and rules to follow are up to each individual sponsor or stakeholder to determine how to set up their standards and rules for defining metadata with the focus on the end user, e.g., programmers and FDA reviewers.

Working with pre-defined Standards, CT/VL Library, Metadata Repository (MDR) Approach

Some companies choose to start defining VLM after database (DB) lock and other companies define all VLM up front before the first patient [is enrolled] in (FPI) a study. There are others who fall in between and may generate study specifications early in the process and save specific define.xml criteria for later. The point in a study when a sponsor chooses to describe clinical trial metadata or study specifications provides another opportunity for establishing rules regarding which variables have VLM and which do not.



The internal operations and standards governance of a company often dictate when and how these processes are done. Setting up pre-defined Standards to use across different therapeutic areas, compounds, and studies takes a great deal of work up front but this will ensure that consistent rules are followed and can save a great deal of time and resources on the backend. By using a Metadata Repository (MDR), setting up a relational database between CT and

VL, or creating a systematic approach to link these concepts has far reaching affects. One of the many values of taking this approach includes creating and re-using consistent metadata across studies.

If links can be provided amongst the different types of metadata then, for example, VLM can automatically be determined once CT is determined. Assume a company has established the standards and the associated metadata for QS as done in the *Tables 5, 6, 7a, 8a, and 8c.* In the QS example, if in the codelist for QSTESTCD a given study only contains Questionnaire A and not Questionnaire B, then we would reduce *Table 7a* to the *New Table7a* below this would automatically establish the values for the *New Table 8a* and the *New Table 8b*.

New Table 7a

CODELIST	SUBMISSION VALUE	DECODE
CL.QSTESTCD	QA1	Questionnaire A1: Is this your first PharmaSUG conference?
CL.QSTESTCD	QA2	Questionnaire A2: Are you a presenter at PharmaSUG 2014

New Table 7a

New Table 8a

VARIABLE	WHERE	DATA TYPE	LENGTH	CODELIST	FORMAT
QSORRES	QSTESTCD = “QA1”	text	1	CL.YN	
QSORRES	QSTESTCD = “QA2”	text	1	CL.YN	

New Table 8a

New Table 8b

CODELIST	SUBMISSION VALUE	DECODE
CL.YN	Y	Yes
CL.YN	N	No

New Table 8b

On a larger scale it is possible to define what is collected within the raw data at the table level and have this information generate the variable level metadata, the VLM, and the controlled terminology. There is great value to stakeholders by using a systematic approach and establishing these relationships. The more that is defined up front and the more pieces of the puzzle that are linked to one another the more systematic and automatic this process becomes. There can be a large reduction in the amount of time and money spent for each study with respect to metadata. Accuracy and consistency increase as these processes become more defined and refined.

The key point is that standards, standards governance, and standards processes should be intrinsic to an organization. How one does things is as important, if not more important, than what one does. The concept of standards is much more than what can be written in an “implementation guide.” Therein lies the true value.

CONCLUSION

Industry standards have moved away from horizontal data seta towards vertical data sets in an attempt to create more standardized structure and provide the capability to use automation, present repeatable structure, and ensure compliance. With the move to vertical standards comes the need to describe the characteristic associated with the values of variables instead of relying on variable naming conventions to convey this information. With few requirements or clear definitions regarding when VLM is appropriate the onus relies on the sponsors or stakeholders to determine the rules. By defining variable relationships an organization can utilize these definitions to help establish consistent rules for implementation. On a larger scale links amongst the various types of metadata can be linked to reduce time and money and to promote consistent standards within an organization thereby adding more value to VLM.

Across the many standards implementation projects d-Wise has worked on, we have found a balance between establishing re-usable internal standards and optimizing process improvement to be the most successful strategy for maximizing efficiencies.

REFERENCES

CDISC Submission Data Standards Team. “Study Data Tabulation Model Implementation Guide.” Version 3.1.3., http://www.cdisc.org/sdtm

CDISC Analysis Data Model Team. “Analysis Data Model (ADaM) Implementation Guide.” Version 1.0, http://www.cdisc.org/adam

CDISC Define-XML Team. “CDISC Define-XML Specifications.” Version 2.0, http://www.cdisc.org/define-xml

EASY FIGURE AND TABLE REFERENCES

Figure 1	Questionnaire CRF Example – used throughout this paper
Table 1	Horizontal data set QUESTION
Table 2	Vertical data set QS
Table 3	Table level metadata for horizontal data set QUESTION
Table 4	Variable level metadata for horizontal data set QUESTION
Table 5	Table level metadata for vertical data set QS
Table 6	Variable level metadata for vertical data set QS
Table 7a	CT for variable level metadata for vertical data set QS – QSTESTCD
Table 7b*	CT for variable level metadata for vertical data set QS – QSTEST
Table 7c	CT for variable level metadata for vertical data set QS – QSORRES
Table 8a	VLM for vertical data set QS
Table 8b	CT for vertical data set QS
Table 8c*	VLM for vertical data set QS – QSTEST

* Blue text = Illustrates QSTEST and the 1:1 relationship with QSTESTCD, choose either *Table 7b* or *Table 8c*
YELLOW highlight = Illustrate options for VLM, choose either “*Table 8a* and *Table 8b*” or “*Table 7c*”

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Name: Shelley Dunn, Senior Life Science Consultant
Enterprise: d-Wise
Address: 1500 Perimeter Park Drive, Suite 150
City, State ZIP: Morrisville, NC 27560
Work Phone: +1 919-334-6089
Fax: 888-563-0931
E-mail: Shelley.Dunn@d-wise.com
Web: www.d-wise.com
Twitter: twitter.com/dWiseTech