PharmaStat℠
drug development consulting and services

CDISC®
Setting the Global Standard for Clinical Data

# Define.xml 2.0: More Functional, More Challenging

CLINICAL DATA INTERCHANGE STANDARDS CONSORTIUM

**Bay Area CDISC Implementation Network**
**24 July 2013**

**John Brega, PharmaStat LLC**

# Presentation Objectives

1. Introduce the new features in Define.xml 2.0 with examples of familiar problems they solve
2. Compare display-formatted documentation from 2.0 with1.0 for a variety of use cases
3. Point out challenges in collecting metadata content for the new features
4. Discuss methods for managing user-editable content

# Background

- Define.xml 2.0 Final was published in March of 2013. It is the first update since CRT-DDS 1.0 was published in 2005.
- 1.0 was designed with SDTM in mind. 2.0 is intended for use with both SDTM and ADaM.
- 2.0 solves several familiar documentation problems that are intractable in 1.0.

# Key New Features in 2.0

## Comments and Document References

1. Comments can now be attached to datasets and method descriptions as well as variables and value-level items.
2. All comments can now include file and page references for hyperlinks to multiple external documents.

# Key New Features in 2.0

## WHERE clauses for value-level metadata

1. Every value-level item is now defined by an associated WHERE clause (declared using a data structure).
2. Any variable can now have associated value-level metadata – not just --TESTCD or PARAMCD.
3. Any value-level item can also have value-level metadata. The structure is recursive.
   (Think you're confused now? Try this…)
4. Enables useful new visualizations of value-level metadata, including "slices".

# Key New Features in 2.0

## Richer Codelist Metadata

1. Identifies codelists based on CDISC (or other) published terminology, indicates if codelist is extensible or not.
2. Differentiates “Enumerated” lists (Mild, Moderate, …) from Code/Decode lists (1 = Mild, 2 = Moderate, …)
3. Identifies values as CDISC or other standard terminology (including sponsor-defined), or as sponsor extensions to standard terminology.
4. Supports greater control over ordering of codelist elements.

# Key New Features in 2.0

## Origins and Data Types (variables and values)

1. Origin now has defined terminology and supports annotated CRF page references for hyperlinks, even to multiple aCRFs.

2. More data types are supported, including partial and incomplete dates/datetimes.

# New Possibilities Come at a Cost

- Some things that used to be impossible are now possible if you can scrounge up the metadata.
- Some things that used to be simple but inflexible are now flexible but complicated.
- 2.0 enables more precise description of your data, but demands more details to achieve it.
- The changes are not incremental or backwardly compatible.  They make current Define documentation systems obsolete.

# Uses for the New Features

1. Add comments to dataset descriptions
2. Use hyperlinks to reference pages within external documents from within Comments on datasets, variables, value items, and methods
3. Describe the values of all variables within a record with a specific --TESTCD or PARAMCD or other common characteristic (a “slice” of data)
4. Connect your codelists to CDISC published codelists and codelist elements

# Use Cases

# 2.0 Solutions

# Examples

# Describing the Content of Datasets

- The Use Case
- 1.0 Limitations, 2.0 Solutions
- Example from 1.0
- Example from 2.0
- New Metadata Needed

# Describing the Content of Datasets

## The Use Case

The table of datasets at the top of the Define cries out for more useful detail. What's *really* in the ZY dataset and why should we care? Were there dataset-level derivations involved?

Some datasets need more explanation than you can put in a dataset label. For these you want to provide a more substantive description.

Sometimes they need *lots* of explanation. For these you want to be able to hyperlink to the relevant pages of an external Study (or Analysis) Data Reviewer's Guide.

# Describing the Content of Datasets

## 1.0 Limitations

- In 1.0 all you have is the dataset label.  If it's the same as in the submitted xpt file, it's only 40 characters.

## 2.0 Solutions

- 2.0 lets you comment on a dataset, and also hyperlink to a page in an external document (like a Study Data Reviewer's Guide) for further description.

# Describing the Content of Datasets

## Example from 1.0

| Dataset | Description | Class | Structure | Purpose | Keys | Location |
|---|---|---|---|---|---|---|
| DM | Demographics | Special Purpose | One record per subject | Tabulation | STUDYID, USUBJID | dm.xpt |
| AE | Adverse Events | Events | One record per adverse event per subject | Tabulation | STUDYID, USUBJID, AEDECOD, AESTDTC, AETERM, AESPID | ae.xpt |
| PE | Physical Examination | Findings | One record per body system or abnormality per visit per subject | Tabulation | STUDYID, USUBJID, PETESTCD, VISITNUM | pe.xpt |
| QS | Questionnaires (SF-36, ECOG) | Findings | One record per questionnaire per question per visit per subject | Tabulation | STUDYID, USUBJID, QSCAT, QSTESTCD, VISITNUM, QSDTC | qs.xpt |

# Describing the Content of Datasets

## Example from 2.0

| Dataset | Description | Class | Structure | Purpose | Keys | Location | Documentation |
|---|---|---|---|---|---|---|---|
| DM | Demographics | SPECIAL PURPOSE | One record per subject | Tabulation | STUDYID, USUBJID | dm.xpt | See Reviewer's Guide, Section 2.1 Demographics<br>Reviewers Guide |
| AE | Adverse Events | EVENTS | One record per adverse event per subject | Tabulation | STUDYID, USUBJID, AEDECOD, AESTDTC | ae.xpt | |
| PE | Physical Examination | FINDINGS | One record per body system or abnormality per visit per subject | Tabulation | STUDYID, USUBJID, PETESTCD, PEDTC | pe.xpt | |
| QSCG | Questionnaire-QSCG (Questionnaires) | FINDINGS | One record per questionnaire per question per visit per subject | Tabulation | STUDYID, USUBJID, QSCAT, QSTESTCD, QSDTC, VISITNUM | qscg.xpt | QS is submitted as a split dataset. The split was done based on QSCAT as QSCG (CLINICAL GLOBAL IMPRESSIONS), QSCS (CORNELL SCALE FOR DEPRESSION INDEMENTIA) and QSMM (MINI MENTAL STATE EXAMINATION). See additional documentation in the Reviewer's Guide, Split Datasets Section.<br>Reviewers Guide |

# Describing the Content of Datasets

## New metadata needed

You need to associate comments with datasets.  No big deal.

Comments used to be just a text string.  Now they're a data structure that can also reference an external document and provide a location reference as a page number or PDF named destination.

The same is true of ComputationalMethod, which is now just called Method.  Used to be text, is now a data structure that can include a Comment, that can include a file reference.

# Describing Value-level Record Types

- Use Cases for SDTM and ADaM
- 1.0 Limitations, 2.0 Solutions
- Variable- and Value-level Examples from 1.0
- Variable- and Value-level Examples from 2.0
- WHERE Clause Example from 2.0
- Data “Slice” Visualization for 2.0
- New Metadata Needed

# Describing Value-level Record Types

## The Use Case for SDTM

In your LB data you have an LBTESTCD of GLUC that is used for both serum and urine glucose. The serum glucose is reported in mg/dL and standardized to mmol/L. The urine glucose is a qualitative dipstick test and therefore unitless.

If your value-level description is tied to LBTESTCD=GLUC, is it referring to the serum test or urine? If serum, is it the reported value in LBORRES or the standardized one in LBSTRESC? How do you specify the codelists for the units?

# Describing Value-level Record Types
## The Use Case for ADaM

The need for useful value-level description in ADaM is both more urgent and more difficult to solve. You often need to document the entire collection of data associated with each record type (PARAMCD). This might include the values of AVAL, AVALC, BASE, CHG, imputed dates, windowed visits, record-level flags, etc.

For a given value of PARAMCD you might need to describe the derivation of all those variables as a group. The story may be complicated enough to justify hyperlinking to a page in the Analysis Data Reviewer's Guide document.

# Describing Value-level Record Types

## 1.0 Limitations

- Value-level metadata describes a variable's values that are associated with a specific value of --TESTCD or PARAMCD. Only one variable is described. Its identity is assumed but not specified.

## 2.0 Solutions

- 2.0 identifies record types with a WHERE clause. Every item with the same WHERE clause belongs to the same record type. With this method you can provide a value-level description of any combination of variables in the dataset.

# Describing Value-level Record Types

## Example from 1.0, part 1: Variable-level Links

| ECG Test Results Dataset (EG) | | | | | | eg.xpt |
|---|---|---|---|---|---|---|
| **Variable** | **Label** | **Type** | **Controlled Terminology** | **Origin** | **Role** | **Comment** |
| STUDYID | Study Identifier | text | STUDYID | Assigned | Identifier | |
| DOMAIN | Domain Abbreviation | text | DOMAIN | Assigned | Identifier | |
| USUBJID | Unique Subject Identifier | text | | Assigned | Identifier | |
| EGSEQ | Sequence Number | integer | | Assigned | Identifier | |
| EGTESTCD | ECG Test or Examination Short Name | text | EG_EGTESTCD | Assigned | Topic | |
| EGTEST | ECG Test or Examination Name | text | EG_EGTEST | Assigned | Synonym Qualifier | |
| EGORRES | Result or Finding in Original Units | text | | CRF Page 21 | Result Qualifier | |
| EGSTRESC | Character Result/Finding in Std Format | text | | Assigned | Result Qualifier | Assigned the same value as EGORRES. |
| EGMETHOD | Method of ECG Test | text | EG_EGMETHOD | Assigned | Record Qualifier | Assigned based on the data collection form name. |

Note: “CRF Page 21” is just a text string. The stylesheet parses it and adds the hyperlink.

# Describing Value-level Record Types

## Example from 1.0, part 2: Value Level

| Value Level Metadata (EG.EGTESTCD~Value) | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Source Variable** | **Value** | **Label** | **Type** | **Controlled Terminology** | **Origin** | **Role** | **Comment** |
| EGTESTCD | INTP | INTERPRETATION | text | EG_EGTESTCD_INTP | CRF Page 21 | | |
| **Value Level Metadata (LB.LBTESTCD~Value)** | | | | | | | |
| **Source Variable** | **Value** | **Label** | **Type** | **Controlled Terminology** | **Origin** | **Role** | **Comment** |
| LBTESTCD | ALB | Albumin | integer | | EDT | | |
| LBTESTCD | ALCOHOL | Alcohol Breath Test | text | LB_LBTESTCD_DRUG | CRF Page 28 | | |
| LBTESTCD | ALP | Alkaline Phosphatase | integer | | EDT | | |
| LBTESTCD | ALT | Alanine Aminotransferase | integer | | EDT | | |
| LBTESTCD | AMPHET | Amphetamines | text | LB_LBTESTCD_DRUG | CRF Page 23 | | |

# Describing Value-level Record Types

## Example from 2.0, part 1: Variable-level Links

**ECG Test Results (EG)** [Location: eg.xpt]

| Variable | Label | Key | Type | Length | Controlled Terms or Format | Origin | Derivation/Comment |
|---|---|---|---|---|---|---|---|
| EGORRES | Result or Finding in Original Units | | text | 15 | | CRF Page 12 | |
| EGORRESU | Original Units | | text | 4 | ["BEATS/MIN" = "Beats per Minute", "msec" = "Millisecond"] <Unit (EGRESU)> | CRF Page 12 | |
| EGSTRESC | Character Result/Finding in Std Format | | text | 15 | | Derived | Data collected in non-standard units is converted using standard conversion factors to standard units. |
| EGSTRESN | Numeric Result/Finding in Standard Units | | float | 5 | | Derived | EGSTRESN = numeric value of EGSTRESC, when EGSTRESC contains numeric data. |
| EGSTRESU | Standard Units | | text | 9 | ["BEATS/MIN" = "Beats per Minute", "msec" = "Millisecond"] <Unit (EGRESU)> | Assigned | |

Note: “CRF Page 12” is a data structure. The stylesheet assembles the display from parts.

# Describing Value-level Record Types

## Example from 2.0, part 2: Value Level

### Value Level Metadata - EG [EGORRES]

| Variable | Where | Type | Length / Display Format | Controlled Terms or Format | Origin | Derivation/Comment |
|---|---|---|---|---|---|---|
| EGORRES | EGTESTCD EQ INTP (Interpretation) | text | 8 | ["ABNORMAL", "NORMAL"] <Interpretation: Original Results> | CRF Page 12 | |
| EGORRES | EGTESTCD EQ PRMEAN (Summary (Mean) PR Duration) | integer | 3 | | CRF Page 12 | |
| EGORRES | EGTESTCD EQ QRSDUR (Summary (Mean) QRS Duration) | integer | 3 | | CRF Page 12 | |

### Value Level Metadata - EG [EGSTRESC]

| Variable | Where | Type | Length / Display Format | Controlled Terms or Format | Origin | Derivation/Comment |
|---|---|---|---|---|---|---|
| EGSTRESC | EGTESTCD EQ QTCB (QTcB - Bazett's Correction Formula) | float | 5.1 | | Derived | QTcB = QT interval / square root of (60 / heart rate). For the complete algorithm see the referenced external document. Complex Algorithms (complexalgorithms.pdf) |
| EGSTRESC | EGTESTCD EQ QTCF (QTcF - Fridericia's Correction Formula) | float | 5.1 | | Derived | QTcF = QT interval / cubic root of (60 / heart rate). For the complete algorithm see the referenced external document. Complex Algorithms (complexalgorithms.pdf) |

# Describing Value-level Record Types

## Example from 2.0, part 3: Use of WHERE for LB

**Value Level Metadata - LB [LBORRES]**

| Variable | Where | Type | Length / Display Format | Controlled Terms or Format | Origin | Derivation/Comment |
|---|---|---|---|---|---|---|
| LBORRES | LBTESTCD EQ BILI (Bilirubin)<br>AND LBCAT EQ CHEMISTRY<br>AND LBSPEC EQ BLOOD | float | 3 | | eDT | |
| LBORRES | LBTESTCD EQ BUN (Blood Urea Nitrogen)<br>AND LBCAT EQ CHEMISTRY<br>AND LBSPEC EQ BLOOD | float | 4 | | eDT | |
| LBORRES | LBTESTCD EQ GLUC (Glucose)<br>AND LBCAT EQ CHEMISTRY<br>AND LBSPEC EQ BLOOD | float | 3 | | eDT | |
| LBORRES | LBTESTCD EQ GLUC (Glucose)<br>AND LBCAT EQ URINALYSIS<br>AND LBSPEC EQ URINE<br>AND LBMETHOD EQ DIPSTICK | text | 8 | | eDT | |

Note: The last two rows illustrate the Glucose example from the SDTM use case.

# Describing Value-level Record Types
## Example from 2.0, part 4: an SDTM "slice"

**Value Level Metadata - LB Slices**

| Test Code | Test Name | Category | Specimen Type | Original Result | Original Units | Standardized Result | Standardized Units |
|---|---|---|---|---|---|---|---|
| BILI | Bilirubin | CHEMISTRY | BLOOD | | ["mg/dL"] | Converted to umol/L. Factor is 17.1. | ["umol/L"] |
| BUN | Blood Urea Nitrogen | CHEMISTRY | BLOOD | | ["mg/dL"] | Converted to mmol/L. Factor is 0.357. | ["mmol/L"] |
| GLUC | Glucose | CHEMISTRY | BLOOD | | ["mg/dL"] | Converted to mmol/L. Factor is 0.0555. | ["mmol/L"] |
| GLUC | Glucose | URINALYSIS | URINE | Results are from a unitless dipstick test. | | Assigned same value as LBORRES. | |

Note: The CDISC stylesheet does not support slices. This example was mocked up in Word.

# Describing Value-level Record Types

## New metadata needed

Easy to say, not so easy to do. The WHERE clause isn't really a WHERE clause. It's a data structure with one or more instances of *Variable Name*, *Operator*, *Literal.* Sorry.

The harder problem is to create a stylesheet or PDF document that assembles the value-level metadata into a useful *and efficient* visualization of record types.

The specification document refers to this problem but the CDISC stylesheet does not include a "slice" visualization. The example I provided was mocked up in Word. Sorry.

# Describing Value-level Record Types

## New metadata needed for Origin

Two of the examples had notes about what was displayed in the Origin column. In 2.0 Origin is a data structure with business rules and controlled terminology for each type of origin. Elements include:

- Type: permissible values are CRF, Derived, Assigned, Protocol, eDT, Predecessor.
- If Type is CRF, provide a document reference, page reference(s), and page reference type.
- If Type is Derived, a Method is required, which may also include a document/page reference.

# Richer Codelist Metadata

- The Use Case
- 1.0 Limitations, 2.0 Solutions
- Example from 1.0
- Example from 2.0
- New Metadata Needed

# Richer Codelist Metadata

## The Use Case

Your LB dataset has 130 unique values of LBTESTCD and LBTEST. Some of those are tests your company invented, the rest should be from CDISC terminology. Either way, each LBTESTCD should come from an authorized list.

How do you track what the authority is for each value of LBTESTCD? Do you know which list it came from and whether it's spelled correctly? Do you have to rediscover those answers each time you review your terminology?

And doesn't it drive you nuts that you have to specify a decode for values that aren't coded to begin with?

# Richer Codelist Metadata

## 1.0 Limitations

- Only one internal codelist type: code/decode.  Most SDTM codelists are just lists of permissible values, which leads to silliness like values decoding themselves.
- No way to document your use of CDISC terminology, sponsor terminology, and sponsor extensions to CDISC terminology.

## 2.0 Solutions

- Added “Enumerated” codelists for permissible value sets.
- Tracks origin and authority of codelists and codelist elements.

# Richer Codelist Metadata

## Example from 1.0

| ACN, Reference Name (ACN) | |
|---|---|
| **Code Value** | **Code Text** |
| DOSE NOT CHANGED | DOSE NOT CHANGED |
| DRUG INTERRUPTED | DRUG INTERRUPTED |
| DRUG WITHDRAWN | DRUG WITHDRAWN |
| **AE_AEREL, Reference Name (AE_AEREL)** | |
| **Code Value** | **Code Text** |
| NOT RELATED | NOT RELATED |
| PROBABLE | PROBABLE |

# Richer Codelist Metadata

## Example from 2.0

### Action Taken with Study Treatment [CL.ACN, *C66767*]

| Permitted Value (Code) |
|---|
| DOSE NOT CHANGED [*C49504*] |
| DOSE REDUCED [*C49505*] |
| DRUG INTERRUPTED [*C49501*] |
| DRUG WITHDRAWN [*C49502*] |

### Causality [CL.AEREL]

| Permitted Value (Code) |
|---|
| NOT RELATED |
| POSSIBLY RELATED |
| RELATED |

### Unit (CM) [CL.CMUNIT, *C71620*]

| Permitted Value (Code) |
|---|
| ug [*C48152*] |
| ug/kg [*C67396*] |
| Other [*] |

* Extended Value

# Richer Codelist Metadata

## New metadata needed

The Codelist structure is a substantial re-do. Many of the new elements are optional but potentially very useful for companies with internal terminology standards.

To use the new elements you may have to carry around a lot of CDISC terminology metadata. The good news is that it can be programmatically validated for referential integrity.

It remains to be seen what integrity checks will be incorporated in OpenCDISC.

# Final Points

# Final Points 1

- Remember that I didn't show *any* actual xml, only stylesheet displays of xml content. Many things that look simple on the screen involve lots of xml elements being collected, interpreted and formatted by the stylesheet.
- The stylesheet in CDISC's distribution package has a clean, updated look (yay!). It is more robust than the 1.0 stylesheets and not disabled by browser security (yay!!).
- The same stylesheet is used for both SDTM and ADaM. Content is displayed differently based on the standard named in the xml document.
- Expect lots of discussion about data slices before we get standard stylesheet support for them.

# Final Points 2

- There are more schema changes than the ones I have mentioned.  Most trade former xml extensions for current ODM schema without changing functionality.
- The net effect is a substantially new and more complex xml structure which obsoletes documentation systems based on 1.0.  Companies with their own systems may have to redesign them top to bottom.

# Final Points 3

- The utility of the new features goes well beyond the use cases and examples shown. *Years* of thought and development have gone into this version.
- The inadequacies of 1.0, especially for ADaM, and the generality and flexibility of 2.0 solutions provide compelling reasons to upgrade.
- In spite of implementation challenges I expect to see general uptake of 2.0 over the next couple of years.

# Thank you!

John Brega:  JBrega@PharmaStat.com