PharmaSUG 2019 - Paper BP-182

Compare and conquer SDTM coding

Phaneendhar Gondesi, TechData Service LLC

ABSTRACT

Appropriate reporting of raw data in CDISC (Clinical Data Interchange Standards Consortium) complaint format is very important in a submission. Mapping and coding are key for reporting. Coding of SDTM (Study Data Tabulation Model) domains could be very challenging especially for a new SDTM programmer. This paper aims to ease SDTM coding by

- Providing general lay out of overall SDTM domain programming
- Similarities in coding process for same variable across different domains
- Differences in coding process for same variable across different domains

INTRODUCTION

This paper focuses on Interventions, Events and Finding classes only. It is assumed that the SDTM programmer uses SAS® software in a windows environment to develop SDTM data sets. Reader should have good understanding of SDTM IG. This paper will follow sections in the same order as in abstract.

GENERAL LAY OUT OF SDTM DOMAIN PROGRAMMING

BEFORE EVEN OPENING SAS EDITOR, HAVE THESE READY

a) Based on study design, decide which domains are needed to map the given raw data.

b) Map Case Report Form (CRF) fields to appropriate SDTM variables.

c) Create specification based on above and SDTM IG (Implementation Guide).

Note: Keep SDTM specifications (specs) and annotated Case Report Form (aCRF) as clear and clean as possible for smoother SDTM Define XML (Extensible Markup Language) creation.

d) Identify key variables for a given domain based on the study design and collected raw data. For e.g., Electrocardiogram (ECG) structure is “One record per ECG observation per time point per visit per subject” per SDTM IG v3.2. But when triplicate ECG measurement are taken at a given time point, afore mentioned structure isn’t enough to identify unique record. Add permissible variable “EGSPID” to key variable list to identify unique record in this situation.

ORDER OF CODING SDTM DOMAINS

e) Create trail design domains first then start working on other key domains like DM, EX, DS, SV, SE and move onto the rest of the domains. This order is based on dependency between the domains.

BEFORE CODING

f) Create macros and formats for variables that are common across different SDTM domains at centralized location. These central macros and formats could help to keep derivation rules and format assignments consistent across different domains within a study or across same domains within a compound. E.g., Epoch assigning macro, Datetime conversion from character to numeric or vice versa.

i) This code automatically reflects the update when the central file is revised, making the coding process more efficient:

```
data frmt;
set central.file;
      fmtname = 'test';
      start = lbtest;
      label = lbtestcd;
      type = 'C';
run;

proc format cntlin = frmt;
run;
```

g) Identify categories of variables.

i) In SDTM IG variables are categorized into different mapping groups based on their

(1) Role: Identifier, Topic, Timing, Qualifier and Rule variables. See section 2.1 SDTM IG v3.2 for details.

(2) Core: Required, Expected and Permissible variables. See section 4.1.1.5 SDTM IG v3.2 for details.

ii) But from a coding perspective, we must identify process involved from raw variables to SDTM variables. Segregate SDTM variables based on similar process and handle them in one segment of the coding. Under coding section, process is mentioned in brackets to give clear idea how this is reflected. Here are different processing methods:

(a) DIRECT: a raw variable that is carried over to SDTM domain variable without any changes except for assigning the CDISC standard label. E.g., STUDYID

(b) RENAME: the raw variable content is not changed but only the variable name and label are modified based on CDISC standards. E.g., VISIT.

(c) STANDARDIZE: converting originally reported raw variables values to CDISC defined units or terminology. E.g., LBTESTCD

(d) REFORMAT: Assigning a format that is different from the originally reported format. E.g., ISO8601 format of datetime variables.

(e) COMBINING: This could sometimes be combining two or more raw variables into one SDTM variable (E.g., PCGRPID) or obtaining data from two or more variables within the same domain. E.g., LBSTRESN for Alcohol and Drug screen data from Clinical Research Unit (CRU) and other Laboratory assessment (Hematology, chemistry etc.) information from external transfer files.

(f) SPLITTING: dividing raw variable content into two or more SDTM variables E.g., COVAL1, COVAL2

(g) DERIVATION: creating a SDTM variable using computations or algorithms. E.g., VSSTRESN for VSTESTCD = BMI using VSORRES values of VSTESTCD in (WEIGHT, HEIGHT).

CODING

h) Create library references for source raw data and SDTM data sets.

i) Clear pre-existing data sets, log and output window:

```
proc datasets lib = work mt = data kill nolist nowarn;
run;

dm 'log;clear;out;clear;';
```

j) Read raw data.

i) Depending on raw data source which could be SAS data sets (like in most CRF data) or external data transfers (Pharmacokinetic or laboratory assessment files), use data step or import/xport procedures respectively. Refer to support.sas.com to identify appropriate data source identifier (DBMS=) under import procedure that corresponds to a specific operating system.

k) Remove any unnecessary formats in the imported data. This will avoid unexpected output or format at the end of programming.

l) Process raw data. Try to code all variables with similar processing (mentioned previously in section g.ii) in the same segment of the program.

i) Keep only the variables that are needed for the specific SDTM data set. This could be direct copy from raw data to SDTM data sets (DIRECT) or raw variables used in creating SDTM variables.

ii) Rename the raw variables if their attributes are different from desired SDTM attributes. This would also avoid any truncation of values while assigning attributes in the later step or prevent losing values when reformatting (RENAME). Simple SQL and datasets procedures as shown below would do the trick:

```
proc sql noprint;
select compress(raw_var_name||"= _"||raw_var_name)
      into: list separated by ' '
from dictionary.columns
where upcase(memname) = 'WANT' and upcase(libname) = 'WORK'
;quit;

%put &list. ;

proc data sets;
      modify want;
      rename &list;
run;
```

iii) Identify variables that need to be re-programmed to CDISC format (STANDARDIZE) E.g., LBTEST, LBTESTCD

iv) For variables that do not fit in above criteria

(a) determine variables that need to be created exclusively within the domain programming. Program these variables in one segment of code. E.g., CMDOSTOT

(b) Use general macros/formats that are created ahead to create the SDTM variables (REFORMAT) E.g., Assigning VSTEST based on raw values.

(c) Create variables by COMBINE or SPLIT processing. E.g., LBCOMMx

m) Merge with other SDTM domains (DERIVATION). This depends on domain created or study design itself. For e.g., LB domain needs to be merged with DM to derive LBDY information. Whereas SV domain is solely based on raw data sources and is not merged with DM. Study reference start date is obtained from raw exposure data set and then SVSTDY or SVENDY are derived.

n) Once we have information from other domains, create variables that are dependent on them (DERIVATION). E.g., timing variables like LBDY or record qualifiers like VSBLFL.

o) Sort by key variables and assign xxSEQ variables based on the key variables for the domain.

p) In the last step, assign renamed variables to the appropriate SDTM variables. Assign attributes to processed data set and output to desired destination folder.

q) Variables for the three general observation classes must be ordered with Identifiers first, followed by Topic, Qualifiers and timing variables. For further details, see SDTM IG v3.2 section 4.1.1.4 for ordering of variables.

Note: In case of data step, use attrib statement following the order of variables of the corresponding domain. Avoid using retain statement for ordering of variables as it could carry over the value from earlier record. Updating the attributes would also be easier when all the attributes are concentrated in one segment of program instead of distributing them across the program.

AFTER CODING

r) Ideally log file should be devoid of any error and warning.

s) Perform a final self qc (even if you are qcing production output)

   i) See if data makes sense like

      (1) Assignment of baseline flags

      (2) Values of derived parameters e.g., BMI

      (3) Expected difference in no. of records in raw vs SDTM data sets.

   ii) Run OpenCdisc software on final domain XPT files for any Pinnacle 21 (P21) errors and warnings.

      (1) If it's an obvious SDTM coding related error then update the program. E.g., Missing one to one correlation between EGTEST and EGTESTCD.

      (2) If it's an SDTM coding related error but need not be fixed then explain in cSDRG. E.g., Variable length is too long for actual data or warning message on certain duplicate records.

      (3) If an error is due to raw data issue then report to Data Manager to resolve it with the site. E.g., Both AESTDTC and AEENDTC have non-missing values but AETERM is missing.

      (4) If an error or warning is from a raw data and

         (a) It cannot be fixed then explain it in the SDTM Reviewers' Guide (cSDRG). These are very individual study specific issues.

         (b) It need not be fixed and it's just the way raw data is, then explain it in cSDRG. E.g., Permissible variables with missing values for all records. PESTAT and PEREASND are missing for all records.

Note: Do not always attempt to modify SDTM coding to bypass P21 error or warning. Maintain traceability from reported raw data to SDTM domain. If reporting would be more transparent then leave P21 error or warning and explain its corresponding reason in cSDRG.

SIMILARITIES IN CODING PROCESS FOR SAME VARIABLE ACROSS DIFFERENT DOMAINS

Please see detailed description of domains in preferred version of SDTM IG. Coding sections mentioned previously, H to K and M to Q are common across most domains in each class. Let's look at the similarities in coding process within variable groups (by role).

Table 1. Similarities in coding process for same variable across different domains

Variable Role	Example variables	Process used	Domain
Identifier	STUDYID	Direct	All events, findings and intervention domain.
	xxSEQ	Derivation	Any events, findings and intervention domain.
	xxSPID	Direct	Relevant events, findings and intervention domain.
Topic	xxTRT, xxTERM	Direct	CM, AE

Variable Role	Example variables	Process used	Domain
Qualifier	xxCAT, xxSCAT	Derivation	Finding domains like LB, VS
	xxDECOD and coding terms like xxBODSYS, xxSOC	Standardize	CM, MH
	xxLOC	Direct	LB, TU
Timing	xxSTDY, xxENDY, xxENRTPT, xxENTPT	Derivation	Events, findings and intervention domains.

DIFFERENCES IN CODING PROCESS FOR SAME VARIABLE ACROSS DIFFERENT DOMAINS

Depending on how data is captured or the domain under question, same variable type could have different coding methods. Below are few examples.

Table 2. Differences in coding process for same variable across different domains

Variable Role	Example variables	Process used.	Domain	Reason
Identifier	xxGRPID	Direct	AE, MH	From CRF
		Derivation	PC, PP	From exposure and visit information.
Timing	EPOCH	Direct (assigned)	SE	Established based on TE.
		Derivation	LB, EG, VS	Obtained by merging with SE and comparing the dates with SESTDTC/SEENDC.

CONCLUSION

Coding of SDTM domains could be difficult for a beginner. Learning the structure of SDTM coding (section 1), recognizing similar trends of coding process between variables (section 2), identifying different coding processes for same variable across different domains (section 3) could help SDTM programmer better visualize the coding process and program SDTM domains more easily.

REFERENCES

CDISC Study Data Tabulation Model (SDTM) v1.4 and Study Data Tabulation Model Implementation Guide (SDTMIG) v3.2. http://www.cdisc.org/sdtm

CONTACT INFORMATION

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

Phaneendhar R Gondesi
TechData Services LLC.
Phanireddy.sas@gmail.com