PharmaSUG 2017 - Paper DS13

Planning to Pool SDTM by Creating and Maintaining a Sponsor-Specific Controlled Terminology Database

Cori Kramer, Ragini Hari, Keith Shusterman, Chiltern

ABSTRACT

When SDTM data are consistently standardized, data can be easily pooled across studies. To consistently standardize raw data to SDTM across studies, there must be an assigned place for each collection field and each collection field must be in its assigned place.

Following the SDTM Implementation Guide and applying Controlled Terminology (CT) as specified is sufficient to support fairly consistent SDTM mapping across studies. To enhance the consistency, sponsor guidelines must be established and maintained.

Establishing, maintaining and enforcing sponsor-defined controlled terminology offers a simple, effective way to ensure that SDTM domains are consistently mapped across studies. We will share an overview of the benefits of leveraging a sponsor-specific controlled terminology database using examples and will suggest how a sponsor-specific CT database can be efficiently developed and used.

INTRODUCTION

While performing SDTM conversion of multiple studies, consistently applying controlled terminology can streamline review and can simplify the process of pooling data for integrated analyses. During SDTM conversion of multiple studies, application of controlled terminology standards is an ongoing process. Therefore, maintaining a database of variable values (CDISC and Sponsor CT) will also need to be ongoing. Below outlines the overall process for compiling such a database.

CREATING THE DATABASE

DETERMINING THE SCOPE OF THE DATABASE

Determining whether standardization will be done across a drug, therapeutic area, or sponsor is the first important step in Controlled Terminology harmonization. This will aide in choosing which variables the CT database needs to cover. Once a sponsor determines the scope of standardization, the database content can be chosen. This list typically will include sponsor-specified CDISC CT extensions (such as additions to LBTEST/LBTESTCD Codelists), CDISC CT additions (such as QNAM/QLABEL), as well as any additional variables that should remain harmonized throughout the lifecycle of similar clinical trials (such as ARM/ARMCD).

CDISC CT extensions are potentially applicable when a variable is associated with an extensible codelist or when otherwise necessary based on the collected data. Sponsor-specified values should remain as consistent as possible so tests can be compared across studies .Every CDISC CT codelist is deemed either extensible or non-extensible. If a given codelist is extensible, then adding extensions to the list is perfectly acceptable within the SDTM model and will not cause any compliance issues. For example, LBTEST and LBTESTCD are both extensible codelists, so adding CT extensions is perfectly valid, given the tests are consistent across studies within scope, and that there are no clear synonyms in CDISC CT. However, it may be necessary to add extensions to even non-extensible codelists at times if required to accurately represent the raw data collection.

CDISC CT additions are variable values that are not associated with a codelist in the CDISC CT version being used. Although there are not usually codelists for values such as QNAM and QLABEL, it is still important to keep the values consistent across studies. An example of this would be coding variables that are not contained in the parent domain (such as MedDRA coding content associated with a concomitant medication indication).

Finally, it is worth considering what other variable values should be consistently standardized across studies. An example of this is ARM and ARMCD. Even if we don’t have exactly the same ARM and ARMCD values across studies, it would be beneficial to include sponsor CT values in the database to ensure that values can be combined across studies. For example, ARM/ARMCD should uniquely define study treatment across studies. Table 1 below is a flawed example. Table 2 is reasonable example.

ARMCD	ARM	STUDYID
A	40 mg Drug A	1001
A	80 mg Drug A	1002
A40	40 MG Drug A	1003

Table 1. Flawed Mapping Strategy

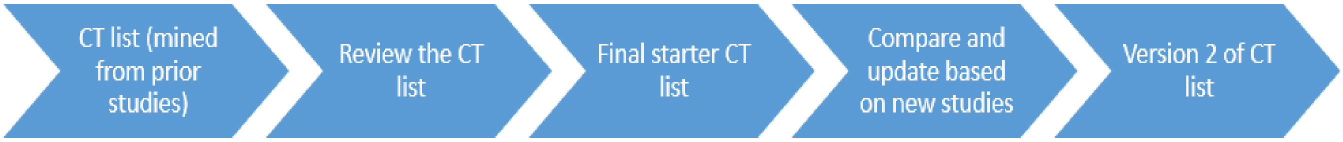
ARMCD	ARM	STUDYID
A40	40 mg Drug A	1001
A80	80 mg Drug A	1002
A40	40 mg Drug A	1003

Table 2. Reasonable Mapping Strategy

METHODS FOR CREATING THE DATABASE

Once SDTM data are present for the first study, simple PROC SQL code can be created to compile unique values for the variables identified.

As an initial step, it is good to mine all unique values (and associated decodes) within identified variables. This step will provide a simple, useful compilation of values that can be used as the basis for evaluation of existing CT use and to determine whether mapping completed on new studies is consistent with the established standard. After harmonization, content from the new studies would be integrated into the CT reference file.


CT list (mined from prior studies)
Review the CT list
Final starter CT list
Compare and update based on new studies
Version 2 of CT list

As a simple example, the below code will provide all unique values of LBTESTCD and the associated LBTEST decodes. This simple approach can easily be generalized for use across variables and domains.

```
PROC SQL;
CREATE TABLE LB_DB AS
SELECT *
    FROM
    (
          SELECT DISTINCT STUDYID, ‘DRUG_NAME’ AS DRUG, ‘THERAPEUTIC_AREA’
          AS TA, LBTEST, LBTESTCD
          FROM STUDY1.LB

          UNION ALL
          SELECT DISTINCT STUDYID, ‘DRUG_NAME’ AS DRUG, ‘THERAPEUTIC_AREA’
          AS TA, LBTEST, LBTESTCD
          FROM STUDY2.LB
    )
ORDER BY STUDYID, DRUG, TA
;
```

Notably, this content will help ensure that sponsor-defined values for tests that are not supported by the LBTEST CDISC Codelist are consistently used.

As an extension of the above example, including additional qualifier variables when mining CT terms and decodes can provide additional value in some circumstances. For example, in labs the same LBTESTCD value can be applicable to more than one discrete type of evaluation and other variables are needed to uniquely define a specific test. These cross variable relations can be used to ensure that both individual variable values and values for related groups of variables are consistently applied. For example, if LBTESTCD = PROT and LBCAT = URINALYSIS, LBSPEC = PLASMA cannot be correct even through PLASMA is and acceptable value for LBSPEC in general. Including all values needed to uniquely define a test can help ensure that CT mapping is consistently applied across variables as well as within a variable. This general approach is the main focus of this paper.

The code below mines unique values of STUDYID, LBTESTCD, LBTEST, LBCAT, LBSCAT, LBMETHOD, LBSPEC, LBORRESU, LBSTRESU.

```
PROC SQL;
CREATE TABLE LB_DB AS
SELECT *
    FROM
    (
          SELECT DISTINCT STUDYID, ‘DRUG_NAME’ AS DRUG, ‘THERAPEUTIC_AREA’
          AS TA, LBTESTCD, LBTEST, LBCAT, LBSCAT, LBMETHOD, LBSPEC,
          LBORRESU, LBSTRESU
          FROM STUDY1.LB

          UNION ALL

          SELECT DISTINCT STUDYID, ‘DRUG_NAME’ AS DRUG, ‘THERAPEUTIC_AREA’
          AS TA, LBTESTCD, LBTEST, LBCAT, LBSCAT, LBMETHOD, LBSPEC,
          LBORRESU, LBSTRESU
          FROM STUDY2.LB
    )
ORDER BY LBTESTCD, LBSTRESU, LBCAT, STUDYID, DRUG, TA
;
```

The DRUG and TA variables can be assigned as above or removed from the code depending if they’re beneficial to have in the central database. Although the above code produces SAS tables, the content would typically be output to an Excel spreadsheet for ease of use.

Once the CT content is available as a database it can be used to programmatically evaluate study data. As one example, off the shelf tools can be used in conjunction with the CT database to help ensure consistent, accurate use of CT. Pinnacle 21 checks can compare define.xml codelists and data. The define.xml also contains full value lists that can help identify inconsistencies. CDISC controlled terminology is built into the Pinnacle 21 validation software, so automated checks are performed to compare value-level information to CDISC terminology by default. Additionally, P21 can compare data with user-defined codelists (such as the sponsor defined CT codelist we have been discussing).

UTILIZING THE DATABASE

The database can be used as a quick reference to locate similar value-level data, to support consistent application of sponsor-defined terminology and may help flag raw data that are not SDTM-friendly.

Table 3 shows a short example of EGTESTCD content. The details would vary based on specific sponsor needs.

STUDYID	Variable	EGTESTCD	EGTEST	ECCAT
ABC-1001	EGTESTCD	INTP	Interpretation	FINDING
ABC-1002	EGTESTCD	INTP	Interpretation	FINDING
ABC-1003	EGTESTCD	INTP	Interpretation	FINDING

Table 3. Sponsor CT Database Example Content

MAINTAINING THE DATABASE

As studies are converted to SDTM, new value-level data will be added to the database by running the existing programs with the new studies added. The code is simple enough that this should be a very minor process, but it can be elaborated upon as can be seen in Appendix 1. Although the programs will need to be rerun to maintain the database, it is important to note that the upfront work creating the programs is only done once, and then new studies can be simply added.

VALUE OF THE DATABASE

There is great value in this process improvement. In the long term, having harmonized controlled terminology makes data easier to review and easier to pool for the purpose of integrated analyses. In the short term, it saves time for programmers by providing a centralized location of previously used SDTM domains and some of their content and saves sponsor resources by streamlining the SDTM mapping process while increasing quality. Table 4 below shows 2 approaches to mapping EGHRMN. The first row is correct, and could be used to support proper mapping. The second row has issues that are highlighted when looking at pooled CT and could have been caught through comparison with row 1 as part of mapping review. While P21 would flag EGTEST as incorrect and inconsistent with EGTESTCD, sponsor terminology is required to ensure that EGCAT is MEASUREMENT when EGTESTCD is EGHRMN as that is a sponsor choice not a CDISC requirement.

STUDYID	Variable	EGTESTCD	EGTEST	EGCAT
ABC-1001	EGTESTCD	EGHRMN	ECG Mean Heart Rate	MEASUREMENT
ABC-1002	EGTESTCD	EGHRMN	Summary (Mean) Heart Rate	

Table 4. Sponsor CT Database – Good and Bad Mapping Example

Identifying data inconsistencies can also show us clear places where CRF databases can be updated for future studies to be as conducive to SDTM mapping as possible. For example, if there are values on the CRF that are inconsistent with non-extensible codelists, these will be identified when compiling the list of values and comparing with the CDISC CT database. In those cases it is worth investigating whether CRF values that match the CDISC CT codelist can be used.

There are a few key ways the database can be used. First, it can be used at a mapping strategy level. For example, if we have a certain test and we're not sure how to map it appropriately, we can easily see if the same test has been used in previous studies and can reference that study for a mapping strategy if the data were collected similarly. The main way that the database will be used is at a value level. When we have a test that doesn't have a CDISC CT value and we can search the database columns for key words that will give us the appropriate value that was used in other studies (if applicable).

CONCLUSION

SDTM data is ultimately more useful when controlled terminology is applied consistently across studies for a given submission (or broader scope if applicable). While setting up a global database does require a bit of work up front, the result is an easy reference to use when applying controlled terminology to new studies within the same scope. The comparison can be facilitated through SAS code and Excel reports, or even handled in the define.xml to allow for automated Pinnacle 21 checks.

ACKNOWLEDGMENTS

We would like to thank Steve Kirby for his review and feedback.

CONTACT INFORMATION

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

Cori Kramer
Chiltern
Cori.Kramer@Chiltern.com

Ragini Hari
Chiltern
Ragini.Hari@Chiltern.com

Keith Shusterman
Chiltern
Keith.Shusterman@Chiltern.com

REFERENCES

[1] Guettner, Joerg; Cuza, Alexandra; ADaM or SDTM? A Comparison of Pooling Strategies for Integrated Analyses in the Age of CDISC, Final, Published PhUSE 2012

[2] Sharma, Rajkumar; Creating an Integrated Summary of Safety Database using CDISC ADaM : Challenges, Tips and Things to Watch Out, Final, Published PharmaSUG 2012

APPENDIX 1

Below code shows a straightforward way to build a sponsor controlled terminology database from existing pooled SDTM domain datasets and to save the CT database to an excel spreadsheet. It is expected that this code would be adjusted to meet your specific needs. Please see the references section for papers that discuss SDTM pooling strategies.

```
/* buildCTDB macro is to create CT database from existing SDTM
datasets */

%macro buildCTDB(in=,out=, outs=, domain=, key=) ;

* Filter all the unique key values from each domain ;
proc sort data=&in out=&out(keep=&key) nodupkey;
  by &key;
run;
```

```
* Filter all the unique key values  from each domain by studyid ;
proc sort data=&in out=&outs(keep=STUDYID &key) nodupkey;
  by STUDYID &key;
run;


%mend;




/* addCT2XL macro will add the CT database to excel spreadsheet using
ODS TAGSETS*/

%macro addCT2XL(in=, domain=);

proc print data=&in;
run;
ods tagsets.excelxp options(sheet_interval='none'
sheet_name="&domain");

%mend;


/* Macro calls to build CT database by each domain */
options spool mlogic mprint;


  %buildCTDB(in=EGPOOL, out=EGCT, outs=EGCTS, domain=EG, key=EGCAT
EGSCAT EGTESTCD EGTEST);

  * Populate the value level metadata for each domain in a separate
sheet in a spreadsheet which will be the first vesion of CT database;

ods tagsets.excelxp file="/SASDATA/cars/dev/ /TACTDB.xls"
style=htmlblue options(sheet_interval='table' );

 %addCT2XL(in=EGCTS, domain=EG); /* CT by STUDYID */
 %addCT2XL(in=EGCT, domain=EGCT); /* Unique CT without STUDYID */


  *ods tagsets.excelxp close;

   %buildCTDB(in=VSPOOL, out=VSCT, outs=VSCTS, domain=VS, key=VSCAT
VSTESTCD VSTEST);

   %addCT2XL(in=VSCTS, domain=VS);
       %addCT2XL(in=VSCT, domain=VSCT);
```

```
     %buildCTDB(in=TAPOOL, out=TACT, outs=TACTS, domain=TA, key=ARM
ARMCD);

   %addCT2XL(in=TACTS, domain=TA);
       %addCT2XL(in=TACT, domain=TACT);


  ods tagsets.excelxp close;
```