

Effective Ways to Manage Thesaurus Dictionaries

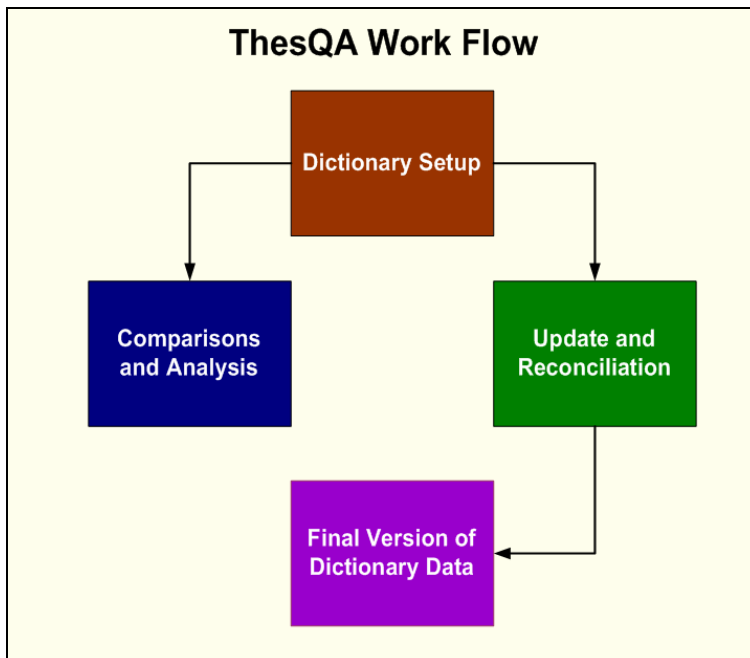
Sy Truong, Meta-Xceed, Inc, Milpitas, CA

ABSTRACT

Coding dictionaries such as MedDRA and WHO Drug can be a challenge to manage with new versions and change control. This becomes even more difficult when different collaborators such as CROs deliver coded data with different coding decisions from various dictionaries. Reconciling these differences can prove to be very resource intensive. This paper will address these challenges and suggest techniques and tools which compare and report on the differences among dictionaries. It will demonstrate strategies on reconciling and managing changes for consistent coding of adverse events, concomitant and medical history.

INTRODUCTION

Coding decisions for adverse events and medications is part science and part art. There is room for interpretation left up to the person deciding on which preferred term or hierarchical System Organ Class (SOC) is associated with the verbatim term. This may differ slightly between projects with different drugs and indications. The difference in coding decisions is compounded when there is more than one person making the decision. This is even further exacerbated when the individuals work in different organizations such as various CROs with different operating procedures. There are many variables contributing to different coding decisions which create a challenge for the data manager who needs to pull all these coding decisions into one coherent and consistent set of coded data for analysis and submission. This paper will describe an approach to manage and reconcile these differences referred to as "ThesQA" or Thesaurus Quality Assurance. The workflow of this methodology is shown here:



The first and pivotal step in the work flow is to be able to manage all the dictionaries centrally by registering them. This is also referred to as “Setup”. Setup gives you the ability to track change control and manage the metadata pertaining to each dictionary. Once you have identified all the versions of dictionaries and their related coding decisions and store the information centrally, you can start to work towards reviewing and reconciling their differences. The goal is to manage all the changes while maintaining change control that takes place during updates.

DICTIONARY SETUP AND MANAGEMENT

The first step in managing your dictionary is to manage the metadata pertaining to each set of data. The metadata is stored in a SAS dataset so that it can be easily updated by SAS tools. An example view of the data would look like:

	Thesaurus Name	Path	Dataset Name	Key Field (1)	Key Field (2)	Key Field	User Name	Date Time
2	MedDRA version 5.0	C:\MedDRA\Data Warehouse	temp.llt	llt_code	llt_name	pt_code	Sy Truong	18NOV05:17:08
3	Who Drug	c:\thesaurus	AUDIT1 WHO	bodysys	preferm	atc_code	Sy Truong	28NOV05:23:22
4	Who Drug 2	c:\thesaurus	WHO23	drugname	genname1	atc_code	Sy Truong	21DEC05:15:11
5	Who Drug 3	c:\thesaurus	WHO22	drugname	genname1	atc_code	Sy Truong	05DEC05:01:55

The SAS dataset named DICTDB, which stands for dictionary database, does not contain the actual values of the dictionary, but rather it captures information about each thesaurus dictionary to be managed. The following steps describe the approach towards setting up the dictionaries.

STEP 1: Identify the types of metadata that are going to be captured.

Dictionary Attribute	Description
Thesaurus Name	This is a unique identifier descriptive name. This name documents what type of dictionary along with the version number.
Storage Path	Path location on the server where the associated data pertaining to the dictionary is stored.
Dictionary Data	Names of all the data storing the dictionary information including verbatim, preferred terms and associated hierarchy classification.
Keys	Key field names used for merging such as verbatim terms, preferred terms and coded classification.

STEP 2: Create a SAS dataset to store this information. The SAS dataset has the following data structure.

Variable	Type	Length	Label
thesname	Char	200	Thesaurus Name
path	Char	200	Path
data	Char	100	Dataset Name
key1	Char	40	Key Field (1)
key2	Char	40	Key Field (2)
key3	Char	40	Key Field (3)
usname	Char	40	User Name
datetime	Num	8	Date Time

The thesaurus name will be used as a unique identifier. The path and data will store the location where the actual data pertaining to the dictionary will be located. The keys will hold the variable names that are identified as keys. The user name and date time will keep track of the last person who has updated the information.

STEP 3: Create a macro which can automate the setup of this setup. The following is an example macro to setup a MedDRA dictionary.

CODE EXAMPLE 1:

```
/*-----*
* Program: tsetup.sas
* Path: C:\Documents and Settings\Sy Truong
* Description: Set up the dictionary MedDRA version 7
*              located at c:\meddra\data warehouse
* By: Sy Truong, %tsetup, 02/05/2006, 1:45:46 am
*-----*/
libname Thesqa "C:\Apache\htdocs\thesqa\data";

%tsetup(thesname = MedDRA version 7,
        path = c:\meddra\data warehouse,
        action = insert,
        data = LLT,
        key1 = llt_code,
        key2 = pt_code,
        key3 = llt_whoart_code);
```

Most of the parameters correspond to the fields that are stored in the dataset. The action parameter however contains the following functions:

- Insert – This will be inserted into the current information as a new row inside the DICTDB dataset.
- Update – This will update the information to an existing dictionary which is already registered.
- Delete – This will delete the information pertaining to the selected thesaurus. Note that it will only delete the metadata and not the underlying data that stores the dictionary.

STEP 4: Create a graphical user interface to capture this same information. This is optional but it can decrease the learning curve of users who are setting up dictionaries for the first time.

ThesQA
Setup Thesaurus Dictionaries

Thesaurus Name:

Path:

Data Sets:

- GLOBALM.BAK
- HLG
- HLG_T_HLT
- HLT
- HLT_PT
- INTL_ORD
- JOE
- LLT**

Action:

- insert
- update
- delete

Key 1: (i.e. verbatim term or drug name)

Key 2: (i.e. preferred term or generic drug name)

Key 3: (i.e. AE preferred term code or ATC code)

Buttons:

In this example, all the information from the macro can be captured through the graphical user interface. The interface also has the feature of being able to generate the macro code once all the selections are made through the “Save Code...” button. All the information captured is therefore stored in the DICTDB dataset which can be previewed through the “View All” button. All updates to the DICTDB will be saved in an audit trail. This captures who has updated the data along with what type of action. This audit trail information can be viewed through the “History” button.

COMPARISONS AND ANALYSIS

There are several types of comparisons that are made during the analysis between the dictionaries. The types of comparisons include:

1. Mismatch – This identifies terms that have been mapped two different ways and therefore are a mismatch.
2. Updated Items – This identifies values that have been updated to the existing dictionary.

The example SAS code segment that is used to perform these comparisons is shown here:

CODE EXAMPLE 2:

```

*** Sort the data by and keep count of number of terms ***;
data &curdat1;
  set &dat1;
  obs1=_N_;
run;

proc sort data=&curdat1;
  by &keyd1;
run;

data &curdat2;
  set &dat2;
  obs2=_N_;
run;

proc sort data=&curdat2;

```

```

by &keyd2;
run;

*** Separate the match and unmatched observations against keys ***;
data _m1 _un1;
  attrib
  thename label='Thesaurus Name 1' length=$200
  thename1 label='Thesaurus Name 2' length=$200
  datname label='Dataset Name 1' length=$200
  datname1 label='Dataset Name 2' length=$200 ;
  merge &curdat1(in=ina) &curdat2(in=inb);
  by &keyd1;
  if (ina and inb) then do;
    datname='&curdat';
    thename='&thes1';
    thename1='&thes2';
    datname1='&newdat';
    output _m1;
  end;
  if (ina and not inb) then do;
    datname="&curdat";
    thename='&thes1';
    output _un1;
  end;
  if (inb and not ina) then do;
    datname="&newdat";
    thename='&thes2';
    output _un1;
  end;
run;

```

In this example, the merge will identify those that contain the same key that has a mismatch. There are several kinds of mismatch. This includes the following conditions:

1. If the data is found with keys in the first data dictionary but not in the second.
2. If the data is found with keys in the second data dictionary but not in the first.

In either case, the mismatch will be captured. Another type of mismatch is when the dictionary has been updated so that there is new data. The following logic will capture this condition.

CODE EXAMPLE 3:

```

*** Get the differences of the observations that have the same key ***;
data _d1;
  set _d1;
  _obs_=_N_;
run;

data _d2;
  set _d2;
  _obs_=_N_;
run;

data _dd1;
  attrib
  thename label='Thesaurus Name' length=$200
  datname label='Dataset Name' length=$200 ;
  set _dd1(drop=_type_ thename);
  thename='&thes1';
  datname='&curdat';
run;

data _dd2;
  attrib

```

```

thename label='Thesaurus Name' length=$200
datname label='Dataset Name' length=$200 ;

set _dd2(drop=_type_ thename);
thename='&thes2';
datname='&newdat';
run;

*** Capture all the same observations ***;
data _m&curdat;
merge _d1(in=ina) _dd1(in=inb);
by _obs_;
if (ina and not inb) then do;
output _m&curdat;
end;
run;

```

In this example, it will check for the number of observations. It will then use this as a key to determine if the value has been updated.

The following steps will describe ThesQA's approach towards implementing this comparison.

STEP 1: Create a macro which captures the information needed to perform these two comparisons.

CODE EXAMPLE 4:

```

/*-----*
* Program: tcompare.sas
* Description: Compare the dictionaries Who Drug v3.1 and Who Drug v3.4
*              by keys: drugname, genname, atc_code.
* Path: C:\Documents and Settings\Sy Truong
* By: Sy Truong, %tcompare, 02/05/2006, 8:34:32 pm
*-----*/
libname Thesqa "C:\thesqa\data";

%tcompare(thes1 = Who Drug,
          thes2 = Who Drug 2);
_obs_=_N_;
run;

```

Once the DICTDB has been defined, all the comparison tool needs to know is the name of the thesaurus dictionaries to be compared. It will then be able to identify the rest of the information to perform the comparisons with the program similar to code example 3 and 5.

STEP 2: A report is generated displaying all the findings from the comparison. Since the findings are captured in SAS datasets, the report can be a simple PROC REPORT with ODS to control the output destination type.

STEP 3: Besides the macro, the ThesQA also contains a graphical user interface to perform the same task.

Similar to the setup interface, the comparisons capture all the parameters of the macro. In addition, it has the ability to generate the macro that is used to perform the comparison from the parameters selected. It also can generate an audit history of the comparisons that have been applied.


UPDATE AND RECONCILIATION

The third and final step in the process is taking the results from the comparison and making a decision. This reconciliation process can include the following types of action.

1. **Add** – In the event that the dictionary data from the first dictionary has been updated and added to the second dictionary, the “add” can be applied. In this case, the data that is found to be the new values will be added to the dictionary that has not been updated.
2. **Drop** – Rather than adding the new updated data, you can choose to drop those values. It could be that the updated information is erroneous and is not correct. In this case, a drop will be applied.
3. **Reconcile** – In the case that a mismatch has occurred during the comparisons, reconciliation can be applied. In this case, the key fields are presented between the source and destination dictionaries. You can therefore choose the correct item to be confirmed.

This is an interactive process. It therefore does require an interactive graphical user interface. The following shows how the process is accomplished.

STEP 1: The user will first decide which two dictionaries are going to be reconciled.

 **ThesQA**
Update Thesaurus Against Another Dictionary

Source:

Path:

Data:

Data label:

Keys:

Destination:

Path:

Data:

Data label:

Keys:

Type: Add Drop Reconcile

In addition to selecting the dictionaries, the type of update will also be selected. There are no decisions to be made for “add” or “drop” but the reconciliation requires the user to decide the term from the source or destination to be selected.

STEP 2 In the reconciliation process, a decision is required to be made. The mismatch term can either be the value from the source or the destination dictionary. The user can therefore decide using the following screen.

ThesQA
Reconcile Mismatch

Source: Who Drug 2

Keys:	drugname	genname1	atc_code
	ACTH	CORTICOTROPIN	H01AA

Destination: Who Drug 3

Keys:	drugname	genname1	atc_code
	ACTH		

Reconciled Selection: 1 of 106958 Mismatch

Source
 Destination

In this case, the values of the keys are presented and users can decide to select either the source or destination items. The destination dictionary will then contain the final decision with all the mismatched values reconciled.

CONCLUSION

You cannot underestimate the amount of resources that goes into coding decisions which can also lead to discrepant dictionaries. This creates challenges which can be alleviated with a strategy, methodology and accompanied tools, referred to as ThesQA. The first step with ThesQA is to manage the metadata to all the dictionaries centrally. This is stored in a dataset and can be managed by the %setup macro. The dictionaries can then be compared to identify mismatches. Mismatches can be resolved by updating the dictionary either by deleting, adding or reconciling mismatches. ThesQA implements a strategy that utilizes logic in the form of a SAS macro and an accompanying graphical user interface to optimally manage thesaurus dictionaries.

REFERENCES

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

ThesQA and all other MXI (Meta-Xceed, Inc.) product names are registered trademarks of Meta-Xceed, Inc. in the USA.

Other brand and product names are registered trademarks or trademarks of their respective companies.

ABOUT THE AUTHOR

Sy Truong is President of MXI (Meta-Xceed, Inc.) They may be contacted at:

Sy Truong
 1751 McCarthy Blvd.
 Milpitas, CA 95035
 (408) 955-9333
 sy.truong@meta-x.com