Paper 108-29

Data Management in Analyzing Clinic Trial Data --- Metadata Application

Julia Zhang, GlaxoSmithKline R&D, Collegeville PA
David Chen, GlaxoSmithKline R&D, King of Prussia PA

ABSTRACT

Efficiently handling Clinical Trial Data based on the Clinical Data Interchange Standards Consortium [1] (CDISC) standard, the regulatory rules from U.S. Food and Drug Administration (FDA) and electronic data submission requirements is very important for Drug Development. Effectively managing Clinical Trial data can vastly increase work efficiency and reduce the probability and introduction of mistakes. The data handling procedure presented here is based on several years of practical working experiences - coupling a metadata application tool on the desktop and specialized SAS macros to apply the metadata to satisfy this need.

INTRODUCTION

As we move forward to an electronic environment, information proliferates rapidly. How do we efficiently manage this voluminous quantity of information and use this data? An appropriate strategy can manage the mass amounts of data.

Descriptive information about an object or resource, whether it is physical or electronic, is called Metadata. It is structured data about data, which allows a piece of information to be discovered by potential users, assessed for its usefulness, used in an appropriate context, and protected, or deleted, as appropriate. For example, the proposed CDISC standard is one type of metadata. Using CDISC standards can improve data quality and accelerate product development in the pharmaceutical industry.

We have arrived in the e-era, where information sources are ever growing, and data spread in the blink of an eye or with a single keystroke. Proper information management is a necessity to ensure that important research is not lost or duplicated unnecessarily. Applying the metadata concept to massive information makes it possible to identify that exact piece of data easily within the ever-expanding sea of information. Without the proper information management practices and tools, the vastness of the information, diversity of the sources, and multiplicity of types can lead to confusion and chaos. Devising and implementing a strategy to efficiently, effectively, and consistently manage data through metadata, and then to utilize the metadata in routine processes, is a key to increase work efficiency in terms of effort and reduce the probability and introduction of mistakes.

INDUSTRY PERSPECTIVE

In the pharmaceutical industry, bringing a new drug to market is the culmination of the drug development process; this means handing off years of accumulated research and analyses for review by a regulatory agency e.g., the FDA, in the form of a submission. In this e-era, the submission has evolved into a paperless, electronic submission. This is an effort to expedite the review process, taking advantage of technological advances of recent years, and due to demands of consumers to bring new, novel, effective drugs to the marketplace. It is also an effort to reduce the number of potential errors associated with manual review within the regulatory agencies. The industry is also under similar pressures to develop efficacious, safe, new drugs and bring them to market as quickly as possible. Both regulatory agency and sponsor benefit from embracing data standards for electronic filing. The agency knows, apriori, what to expect in any submission package i.e., structure and content, and the sponsor has specific guidelines to follow across all filings.

The establishment of global, vendor-neutral, platform independent standards for the medical and biopharmaceutical industry, leading to improved data quality and acceleration of product development, is the mission of the Clinical Data Interchange Standards Consortium. CDISC is committed to the development of industry wide standards to support the electronic acquisition, exchange, submission and archiving of clinical trial data and metadata. Of particular relevance are the Submissions Data Standards Model [2] (SDS) and the Analysis Data Models [3] (ADaM) and their underlying metadata specifications. That having been said, implementation and integration of the metadata elements during the development of the data domains and analysis data sets specified in the SDS and ADaM, respectively, is the focus of this paper.

DATA HANDLING PROCEDURE

Efficiently handling data leads to increased work efficiency and reduces the probability of the introduction errors. It will also accelerate the Quality Control (QC) process and enhance the quality through the application of metadata.

The following diagram (Figure 1) summarizes a data handling procedure. Some SAS macros (exist.sas and cleanup.sas) and a metadata application tool [4] were developed to implement the CDISC elements and satisfy the FDA submission requirements. The details of this procedure will be explained in the following paragraphs.

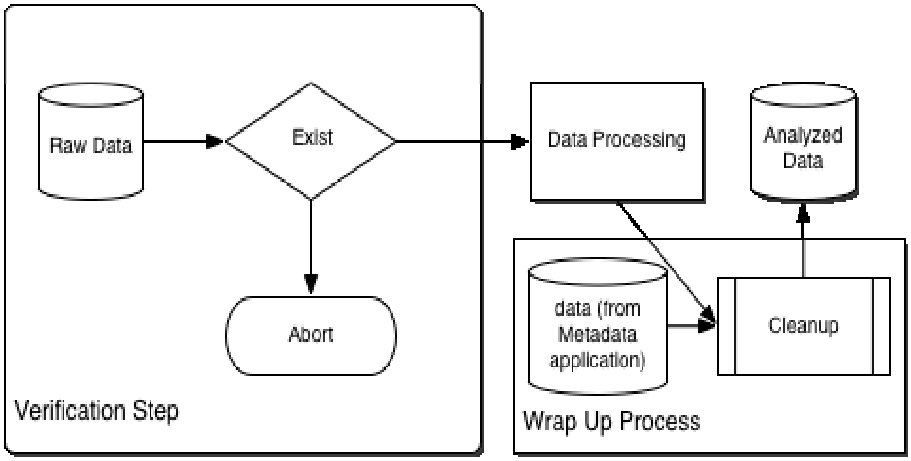

Figure 1. Flow chart for data handling.

The integration and implementation of CDISC's SDS and ADaM models begins with a metadata specification of all the data domains and analysis data sets for the study. This Data Definition Table (DDT), as well as other study reporting related resources, is centralized in an Excel workbook. It provides an effective means to manage the content and structure of data domain and analysis data sets. The idea is to utilize the metadata described in the DDT to maintain consistency of these elements, from specification to implementation, across studies during the data analysis and processing phases. A metadata application tool [4] was developed to meet this need. Figure 2 is an example of a display from the developed metadata application tool.

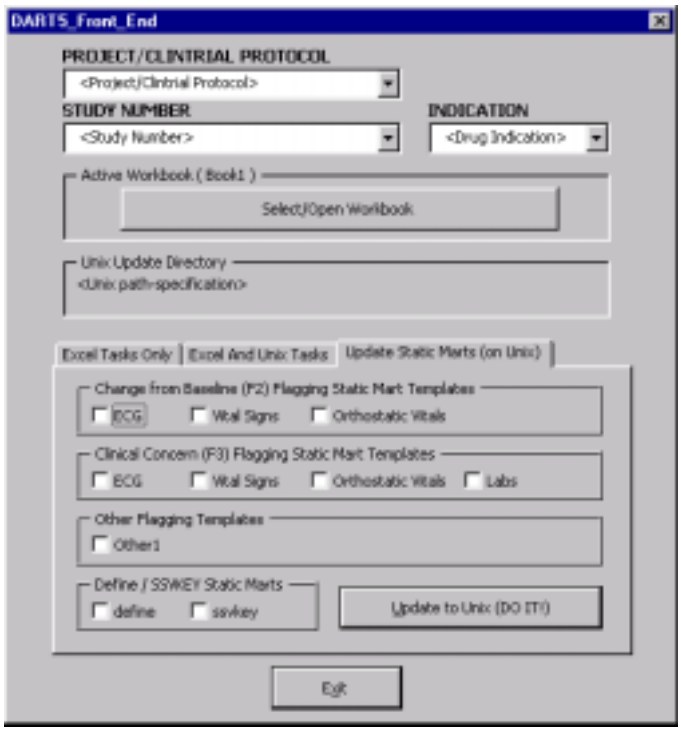

Figure 2. The display of the developed metadata application.

The metadata application tool serves several functions - among them providing the link between the DDT, which resides on the desktop, and the target-reporting platform. Though the link is not dynamic – it does transfer the metadata onto the development and reporting platform. The metadata is in the form of a SAS data set and is then referenced, most notably, in the SAS programs used to create the data domains and analysis data. Each data set utilizes the metadata from the DDT and follows the general steps of:

- verifying the existence of necessary resources;
- derivation of variables per the specification;
- cleaning up the entire process by applying the metadata from the DDT (Figure 1).

The verification step – first step in Figure 1, ensures that the required raw data sets and user-defined formats exist prior to continuing with the data processing. This allows the user to maintain control over the program so that the procedure continues only if the required components are present. A SAS macro, exist.sas, handles this task. This macro checks for the existence of needed formats and input data sets using the SAS cexist() and exist() functions - note that the cexist() function is being called within another SAS macro which will search the catalogs in the order defined by the fmtsearch= system option. If any one of the required formats or input data sets are not found, processing of the SAS session is terminated in a controlled, forgiving manner by the ABORT ABEND *n* statement. All items (i.e., input data sets and formats) not located by exist.sas will be printed to the SAS log. The following example shows some core parts of the exist.sas macro. Note, it is not a complete macro.

```
%macro _exist(_input=,
              _format=,
              _module= UNKNOWN);

/*************************************************************************/
/*   Declare local macro variables                                       */
/*     index- loop index counter                                         */
/*     listitem- item of input dataset or _format                        */
/*     notfound- flag with status of existence (0:non-existence, 1:existence  */
/*************************************************************************/
  %local index
         listitem
         notfound;

  %let notfound= 0;
/*************************************************************************/
/*   Only continue execution when the dataset list is not empty          */
/*************************************************************************/
  %if (&input &format ~= %str( )) %then
    %do;
      data _null_;

/*************************************************************************/
/*   Verify the existence of the datasets in the _input parameter list.  */
/*   Only continue if the list is not empty.                             */
/*                                                                       */
/*   Initialize the loop counter index.                                  */
/*   Get the item from the dataset list.                                 */
/*************************************************************************/
        %if (&_input ~= %str( )) %then
          %do;
            %let index= 1;
            %let listitem= %upcase(%scan(&_input, &index,%str( )));

/*************************************************************************/
/*   Loop thru each item in the list until the dataset list is exhausted.*/
/*************************************************************************/
            %do %while (&listitem ~= %str( ));

              %if ( not(%sysfunc(exist(&listitem))) ) %then
                %do;

/*************************************************************************/
/*   The dataset does not exist.                                         */
/*                                                                       */
/*   If this was the first time existence criteria has failed for this module */
/*   go to a new page and generate the failure heading.                  */
/*                                                                       */
/*   Write out a message indicating dataset not found.                   */
/*   Set the flag to failure ie., notfound= 1                            */
/*************************************************************************/
               %if (&notfound = 0) %then
                %do;
                  put _page_;
                  put ">>>>>>>>> EXISTENCE CRITERIA FAILURE IN MODULE - &_module"
                %end;
                  put ">>>>>>>>>>>       ERROR: DATASET (&listitem) NOT FOUND." ;
               %let notfound= 1;
               %end;
```

```
/*************************************************************************/
/*   Increment loop index.                                               */
/*   Attempt to get next item from the list.                             */
/*************************************************************************/
             %let index= %eval(&index + 1);
             %let listitem= %upcase(%scan(&_input, &index,%str( )));
           %end;

           %if (&notfound) %then
              %do;
                put ">>>>>>>>>>>";
              %end;
         %end;

/*************************************************************************/
/*Verify the existence of the formats using the same strategy which is   */
/*excluded here                                                          */
/*************************************************************************/
/*Only if at least one item from the list was not found the abort the program */
/*************************************************************************/
        %if (&notfound) %then
          %do;
            abort abend 68;
          %end;
    %end;

%mend _exist;
```

Data manipulation/derivation of the variables specified in the DDT follows the existence step (Figure 1). This will depend on the programming styles and the needs of the particular study. Though not implemented, this step could conceivably reference the metadata specification for the data domain or analysis data set being generated and obtain the derivations directly from the metadata- if it had been described appropriately (i.e., as SAS code). Are there benefits to doing this? Certainly there is the potential reduction of effort since the derivation only needs to be specified once- in the DDT. However, many derivations, especially for analysis data sets, can be complex, multi-step endeavors involving several variables and relying on intermediate results- as such, this option was not considered practical and was abandoned. Another factor contributing to this decision was the fact that there is some measure of efficiency gained by aggregating several variable derivations within a single data step - a benefit lost if each variable were to be derived independently. The choice is, of course, subjective; nevertheless, it is possible to apply the metadata in the derivation step.

Having derived the necessary variables, the focus shifts to ensuring the data set has the structure and content as described in the DDT. Having the metadata available on the target platform means not having to duplicate effort by re-specifying the attributes of the data domain and analysis data sets. During the data processing step, perhaps you forgot to delete some temporary variables or perhaps didn't fully specify the attributes of each and every variable on your data set- but this really isn't a problem if you apply the metadata. Of course, there is no remedy for incorrectly derived variables, but assuming the derivations are correct and complete. It is possible to apply the metadata and arrive at a final data set having the structure and attributes of the DDT in this wrap up step (Figure 1).

The metadata for the data domain and analysis data sets include the data set name, the data set label, the variable name, the variable label, the variable type, the variable length, the format associated with the variable (e.g., for date or time variables), a flag specifying whether the variable should be hidden and the desired sort order for the data set. Applying the metadata to the data set is handled by a SAS macro, cleanup.sas, - the cleanup step in Figure 1. The cleanup macro extracts the metadata for the appropriate data domain or analysis data set from the metadata SAS data set version of the DDT, which has been made available with the metadata application described earlier. It performs the following tasks:

- verifies the existence of the metadata data set prior to proceeding;
- suppresses variables from the final data set that are identified as hide variables, if requested;
- creates final data set containing only the variables defined by the metadata;
- attaches the data set label to the data set as defined by the metadata;
- applies the type and length attributes to the variables as defined by the metadata;
- applies the labels to the variables as defined by the metadata;
- sorts the data set according to the order specified as defined by the metadata;
- prints sample observations from the final data domain or analysis data set to list.

The resulting data set matches the specification of the DDT in structure and content exactly. How does this reduce effort and increase efficiency during QC? Since the data set has been created with the metadata as specified in the DDT- the document that is supplied to the FDA, it is possible to essentially skip the process of verifying that the data set reflects the specification. Granted, there are only marginal time savings for one data set but as the number of

data sets increases, the time savings are more substantial. This use of metadata also saves time during coding since it's no longer necessary to type in, at the minimum, the data set and variable labels. Here again, a fast typist may not see any significant time savings with one data set but across several data sets the reduction in effort becomes apparent- especially considering it's only necessary to type these once, in the DDT.

The wrap up process is used during the creation of the SAS transport data sets to comply with the Title 21 Code of Federal Regulations (21 CFR part 11) Electronic Records; Electronic Signatures guidance [5]. Data domain and analysis data sets are required elements to an electronic filing but these data sets as used internally to generate reports may contain variables to facilitate reporting (i.e., coded variables, ordering variables) but not required by the regulatory agency for their review. As mentioned earlier, there is a hide flag in the DDT indicating whether the variable is to be "hidden" or not. The cleanup macro is written so that the variables having the hide flag attribute selected are excluded from the resulting data set- quite handy if you consider all it takes is calling cleanup.

CONCLUSION

This paper suggests an efficient procedure for data handling in the creation of data domain and analysis data sets, and an appropriate use of metadata. Applied wisely, metadata can circumvent many chaotic situations in data handling and increase work efficiency by reducing the time necessary to accomplish certain tasks by automating/standardizing them and eliminating duplication effort. In the drug development process – satisfying the electronic submission requirements of regulatory agencies and following the guidelines proposed by CDISC are intricately linked. Capitalizing on the use of metadata by integrating it into the data domain and analysis data set creation process is one strategy that can lead to a reduction of time and effort, and a guarantee of a certain level of quality in our electronic submissions.

REFERENCES

(1) CDISC, Standards, http://www.cdisc.org/standards.
(2) CDISC, Submissions Data Model (SDM) Version 2.0, http://www.cdisc.org/models/sds/v2.0.
(3) CDISC, Analysis Dataset Model, http://www.cdisc.org/models/adam/ADaM_Guidelines_V1.pdf
(4) Zhang, J., Chen, D., and Wong, TL, Metadata Application on Clinical Trial Data in Drug Development, SUGI28, paper 238-28, 2003.
(5) FDA Title 21 Code of Federal Regulations (21 CFR Part 11) Electronic Records; Electronic Signatures, http://www.fda.gov/cder/gmp/index.htm.

TRADEMARKS

CONTACT INFORMATION

Julia Zhang
GlaxoSmithKline
1250 South Collegeville Road
Collegeville, PA 19426
Work Phone: 610-917-6914
Email: julia.z.zhang@gsk.com

David Chen
GlaxoSmithKline R&D
2301 Renaissance Blvd.
King of Prussia, PA 19406-2772
Work Phone: 610-787-3828
Email: david.c.chen@gsk.com