Paper CD11

ADaM on a Diet: Preventing Wide and Heavy Analysis Datasets

Van Krunckelsven Dirk, Merck Serono S.A., Geneva, Switzerland

ABSTRACT

One of the problems with CDISC's Analysis Data Model (ADaM) is the so called « analysis-enabling » variables. Often all possible subgroups, stratifiers, baseline values, categories of baseline values, etc. are all created in the Subject Level analysis dataset (ADSL). Frequently, these are all merged onto every other dataset. Accommodating the potential need for outputs or exploration.

An alternative is proposed that limits the width of the ADSL dataset: storing such additional information in a BDS-type structure. The principle is similar to the Supplemental-feature in SDTM. Provided that parameter value level metadata is available, transposing and merging such supplemental subject-level information onto other ADaM datasets is trivial.

This approach enables standardized storage of subject level information not defined in CDISC's ADaM. It keeps ADSL slim and all other datasets focused. However, all information remains readily available for additional analyses on the ADaM – and potentially the SDTM – data.

THE ANALYSIS DATA MODEL (ADaM)

With the December 2009 release of the Analysis Data Model Implementation Guide (ADaMIG) v1.0, CDISC also upversioned the Analysis Data Model (ADaM) to 2.1. These releases greatly improved both the data models and the documentation: clear models and rules are now available. In ADaM two types of analysis data structures are being described: the Subject Level Analysis Dataset (ADSL), and the Basic Data Structure (BDS). The ADSL contains "variables that describe attributes of a subject" (ADaMIG, page 8). The BDS is a semi-normalized data structure that contains "key endpoints and data that vary over time during the course of a study" (ADaMIG, page 9). Since this release in early 2011, CDISC has also provided a draft ADaM Data Structure for Adverse Event Analysis (ADAE), and a draft Basic Data Structure for Time-to-Event analysis (ADTTE).

WIDE AND HEAVY ANALYSIS DATASETS

In analyses additional variables are often required such as subgroups, baseline values and categories of baseline values. Within the context of an ADaM-based set of analysis datasets, the most obvious place for such data is the ADSL. Indeed, the ADaMIG states that "ADSL is a source for subject-level variables used in other analysis datasets". Such variables can be considered "Analysis-Enabling", the ADaMIG considers stratification and subgrouping variables as examples thereof. ADaM mentions these as belonging to ADSL.

ADaM, however, also stresses that only fields relevant to the analysis datasets should be included. "The inclusion of too many extraneous variables […] makes it more difficult for users to find important variables and can impede clear and concise communication." (ADaM, page 28)

Practice, however, shows that often subgroups, stratifiers, baseline values, categories of baseline values, etc. end up on the ADSL dataset and are then copied over to most, if not all, of the other analysis datasets. Often comments like "If your variable is important enough to be on the ADSL, then it should be present throughout all of the datasets" are heard. This leads exactly to unclear datasets that are wide and heavy. In this paper, a mechanism is proposed to prevent this data obesity.

ADSLSUPP: SUPPLEMENTAL SUBJECT LEVEL INFORMATION

The solution that is being explored here is to concentrate all such subject-level "analysis-enabling" variables into an additional dataset: ADSLSUPP. This dataset is then to contain all subject level information that is not on the ADSL model as described by CDISC. The principle is very similar to that of the supplemental qualifiers existing in the Study Data Tabulation Model (SDTM) but in an ADaM style, i.e. based on the BDS.

ADSLSUPP

Table 1 describes the standard variables used in ADSLSUPP. The information is mainly a copy from what is available in the ADaMIG. There are, however, three important differences. These are described further down and italicized in the table. The Core column has the exact same meaning as in CDISC's ADaMIG: Req = Required, the variable must be included on the dataset; Cond =Conditionally Required, the variable must be included under certain circumstances; Perm = Permissible, the variable may be included but is not required. Note that ADaM has a different definition for "Required" than SDTM. In SDTM these variables cannot be null for any record, in ADaM this is not the case.

Variable Name	Variable Label	Type	CodeList / Controlled Terms	Core	Notes
STUDYID	Study Identifier	Char		Req	SDTM DM.STUDYID
USUBJID	Unique Subject Identifier	Char		Req	SDTM DM.USUBJID
PARAM	Parameter	Char		Req	The description of the analysis parameter. *The length cannot exceed more than 40 characters.*
PARAMCD	Parameter Code	Char		Req	The short name of the analysis parameter in PARAM. The length cannot exceed more than 8 characters.
PARAMN	Parameter (N)	Num		Perm	Useful for ordering and programmatic manipulation.
PARAMTYP	Parameter Type	Char	(PARAMTYP)	Perm	Indicator of whether the parameter is derived as a function of one or more other parameters. This should not be confused with DTYPE which is relevant to derived AVAL and/or AVALC values.
AVAL	Analysis Value	Num		Req (at least 1)	Numeric analysis value described by PARAM. *AVAL must not be filled in for records that have AVALC filled and vice versa.*
AVALC	Analysis Value (C)	Char			Character analysis value described by PARAM. *AVALC must not be filled in for records that have AVAL filled and vice versa.*
DTYPE	Derivation Type	Char	(DTYPE)	Cond	Analysis value derivation method. DTYPE is used to denote, and is required to be populated, when the value of AVAL or AVALC (and thus the entire record) has been imputed, derived, or copied from other record(s). DTYPE is required to be populated even if AVAL and AVALC are null on the derived record. DTYPE is not used to denote that an analysis parameter is derived. PARAMTYP may be used to indicate that an entire parameter is derived. For each value of DTYPE, the precise derivation algorithm must be defined in analysis variable metadata, even for DTYPE values in the controlled terminology.
NUMCHAR	*Numeric or Character*	*Char*	*(NUMCHAR: N C)*	*Perm*	*Contains N for Numeric Parameters, C for Character Parameters. Clearly identifies whether the value for the parameter is contained in AVAL (for N values of NUMCHAR) or in AVALC (for C values of NUMCHAR). The same value for the same Parameter.*
DECP	*Number of Decimal Places*	*Num*		*Perm*	*Contains an integer indicating the number of decimal places available for the value. The same for the same Parameter.*
CHARLEN	*Length of Character String*	*Num*		*Perm*	*Contains an integer indicating the length of the character string. The same for the same Parameter.*

Table 1 : Variables in ADSLSUPP, Supplemental Subject Level Analysis Dataset.

A first difference with the information provided in the ADaMIG is the maximum length of the content of the PARAM variable. Here, in ADSLSUPP, it is restricted to 40 characters. The goal is to easily remerge these variables wherever needed onto other datasets. In this case the text provided in PARAM will become the label of the variable created, that is why the limitation kicks in for ADSLSUPP, whereas the ADaMIG imposes no such restriction for BDS.

Note also the second difference which is for both AVAL and AVALC. These variables are not to be filled in at the same time. Per parameter, only one variable will be created and merged onto the target dataset: it is either numeric or character. If both AVAL and AVALC are filled for a parameter, it is not clear which one is to be merged, therefore only one of them can be filled in.

Those familiar with the BDS will also have noticed the third difference: The permissible variables NUMCHAR, DECP and CHARLEN. These variables do not exist on the BDS. The Notes column describes their purpose. The reason for including these variables on the proposed ADSLSUPP structure is being described further down in the section Parameter Value Level Metadata is Vital.

REMERGING ADSLSUPP IS TRIVIAL AS THE KEY IS USUBJID

Because ADSLSUPP only contains information at the level of the subject, remerging it onto any other dataset is trivial because the key will always be USUBJID. There is no need to check out the identifying variable and its value as is the case with the supplemental qualifiers in SDTM with IDVAR and IDVARVAL. There is no danger for identifying variables that do not exist. The point is that the nature of the data collected in ADSLSUPP makes it straightforward to plug it onto other data in a far more intuitive way than is the case for SDTM’s supplemental qualifiers.

The IDVAR and IDVARVAL principle is not overly difficult, nevertheless quite some resistance against it has been observed; the main argument always being that it complicates matters. For ADSLSUPP the remerge is a piece of cake as is shown below where the ADAE data is merged with the numeric FSH Baseline Value (AVAL). The below is based on proc transpose and the obligatory sorts. One can obviously just as easily join the data using proc sql or using datasteps avoiding the costly transpose procedure altogether. The whole code can be generalized easily into a fairly simple macro. Describing such a macro is, however, not in scope of this paper.

```
proc sort data=adae out=adae_srt;
  by usubjid;
run;

proc sort data=adslsupp (where=(PARAMCD=’FSHBL’)) out=adslsupp_srt;
  by usubjid;
run;

proc transpose data=adslsupp_srt out=adslsuppT (keep = usubjid FSHBL);
  by usubjid;
  var aval;
  id paramcd;
  idlabel param;
run;

data adaeBL;
  merge adae_srt adslsuppT;
  by usubjid;
run;
```

The example shown is for SAS; other software will handle it just as easily.

PARAMETER VALUE LEVEL METADATA IS VITAL

As is the case with any BDS type dataset in ADaM: Parameter Value Level Metadata is vital. It describes the parameter in further detail. What type of data does it contain: character or numeric? To what level of precision is the information available? Is there any controlled terminology at this level?

Without this additional information, the data become hard to interpret. In the code provided above, AVAL is underlined because this is the unknown. How does one know that the Baseline Value for FSH that is being merged onto the AE analysis data is numeric or character and thus sits in the AVAL or the AVALC variable? One knows only because it was said to be numeric in the text above the code (also underlined).

In the proposed ADSLSUPP structure it is a must to determine whether AVAL or AVALC contains the value for the parameter at hand. Whether it is numeric or character data is thus vital. Because of this, the structure for ADSLSUPP shows the NUMCHAR variable indicating exactly this. Indeed, this creates some redundancy as for a given value of PARAMCD the value of NUMCHAR must always be the same. Such information is expected to be present in the data definition specification, the define.xml file. Despite all of this, the author prefers the information to sit alongside the actual data because:

a) a define.xml file is not necessarily available
b) not everyone reads such a define.xml file very easily to determine the necessary information
c) having the information in yet another location adds an additional layer to the merge

The proposed structure including the Permissible variables NUMCHAR, DECP and CHARLEN is fully enclosing all the information required to remerge the data onto any other data that has the right key available (USUBJID). NUMCHAR, DECP and CHARLEN are set to Permissible because they do not exist in either the ADSL or BDS model. In the author’s implementation they would be mandatory. In fact, the author would like to see the same or similar information on the BDS as well as the SDTM findings domains and supplemental qualifier structures.

DISCUSSION TOPICS

USAGE: REDUNDANCY?

How to use this ADSLSUPP principle? When an output requires an ADSLSUPP variable, it is merged onto the relevant dataset, e.g. ADAE, so that the output can be created. Then what: Do you leave the variable on ADAE and have it in ADSLSUPP thus creating redundancy? This way the analysis readiness is maintained while the data points are still available in ADSLSUPP for further analyses.

ADSLSUPP, THAT’S AN UGLY NAME!

The dataset name is based on the ADaM requirement to start with the prefix “AD”. Next it indicates that the information is Subject Level (SL) as well as Supplemental (SUPP), therefore ADSLSUPPan alternative would be BDSL for Basic Data Subject-Level, but what’s in a name…

BASIS: BDS OR SUPPQUAL

The underlying paper uses the BDS as the basis for the structure of the Supplemental Subject Level Analysis Dataset. An alternative would be to base the structure on the Supplemental Qualifier structure as described in the SDTM. Both ways are viable, the BDS route was chosen over the SUPPQUAL structure as numeric data is then immediately available as such without requiring further transformation. Also IDVAR and IDVARVAL are not required for subject-level information, the key is always USUBJID.

DATA DEFINITION IN THE SAME STRUCTURE

Some people may not like having the data definition sit alongside the actual data, therefore, in this ADSLSUPP structure they are set to permissible. In the end, when the data are being submitted, the define.xml would be available. Others may like the principle recognizing that in the end a define.xml file is needed, but all along the chain where people work on and with the data, the inclusion may be beneficial.

Some may even argue that the structure needs adaptation so that one can also define dates, times and datetimes in ADSLSUPP. In that case it is probably best to rename CHARNUM to be more general, like AVALTYP allowing for an extended set of controlled terms that would cover all of the cases: character, numeric, date, time, datetime. The NUMCHAR row in Table 1 can then be replaced by what is described in Table 2:

AVALTYP	*Analysis Value Type*	*Char*	*(NUMCHAR: N C D T DT)*	*Perm*	*Contains N for Numeric Parameters, C for Character Parameters. D for Dates, T for Times, DT for Datetimes. Clearly identifies whether the value for the parameter is contained in AVAL (for N, D, T and DT values of NUMCHAR) or in AVALC (for C values of NUMCHAR). The same value for the same Parameter.*

Table 2 : Extending the data definition variable. NUMCHAR becomes AVALTYP allowing more data types.

REVIEWER ACCEPTANCE

The hottest discussion point: acceptance from the community, but more importantly from the reviewers, is needed. Will the reviewers accept a method that does not provide every piece of information readily available within an analysis dataset? Will they prefer this method over wide and heavy analysis datasets? Or is the proposed structure just a working tool that allows sponsors to work more easily with the data before creating a final set of analysis datasets for submission?

SDTM FOR TABULATION OF ENHANCED COLLECTED DATA AND ADAM FOR DERIVATION AND ANALYSIS

In general, the impression is that the reviewers have not yet fully embraced ADaM as the data source for analysis data. All the talk is about SDTM. Amendment 1 to the SDTM, for example, (still in draft at the time of writing) requests a treatment emergence flag for AEs. Moreover it requires this flag to be identical as the flag used for the analysis. The same draft document also introduces a death flag onto the SDTM DM domain. Such items, the author would have expected to appear in analysis datasets, not in tabulation datasets. Nevertheless, despite being in draft status this amendment is referenced by CDER in their Common Data Standards Issues Document.

Is SDTM an enhanced version of the collected data or is it leaning more and more towards analysis data? A choice must be made. Is SDTM the source for analysis or is ADaM?

CONCLUSION

The underlying paper pinpoints a potential issue in ADaM: datasets easily grow too wide, which puts at risk a goal of the Analysis Data Model, namely clear and concise communication. A potential solution is being proposed: put “less important” variables together in a supplemental-like structure and merge-on only as required. This proposal has some clear advantages: (1) all required output can be produced, (2) reviewers can merge on any of these supplemental subject level variables as they please to perform their own analyses, (3) all data can be captured in a standard form.

ADSLSUPP is, however, merely a proposal. Community and reviewer acceptance is crucial. In presenting this paper, it is hoped that ADaM will become a discussion topic more frequently. A lot remains to be discussed about standard data and metadata. Hopefully this contributes.

REFERENCES

CDISC Analysis Data Model Version 2.1.

CDISC ADaM Implementation Guide Version 1.0.

CDISC Study Data Tabulation Model Version 1.2.

CDISC SDTM Implementation Guide Version 3.1.2.

Draft Amendment 1 to SDTM v1.2 and SDTMIG v3.1.2.

CDER Common Data Standards Issues Document Version 1.0 / May 2011.

CONTACT INFORMATION

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

Dirk Van Krunckelsven
Merck Serono
9, Chemin des Mines
CP 54
CH-1211 Geneva 20
Switzerland
Email: dirk.van.krunckelsven@merckgroup.com
Web: http://www.merckserono.com