PhUSE 2009

Paper CD03

An Evaluation of the ADaM Implementation Guide v1.0 and the Analysis Data Model v2.1

Chris Price, Roche Products Ltd, Welwyn Garden City, UK

ABSTRACT

In June 2008 the ADaM Implementation Guide version 1.0 (ADaMIG v1.0) Draft for Public Comment and the Analysis Data Model v2.1 (ADaM v2.1) Draft for Public Comment were released by the CDISC© Analysis Data Model Team. This paper will evaluate the high level concepts explained in these two documents and explore the challenges encountered when implementing these guidelines in a real life implementation. This will be compared and contrasted with an existing internal analysis dataset standard and will explore areas where I feel ADaM needs to be enhanced and extended to allow it to move forward in gaining widespread acceptance within the pharmaceutical industry. A knowledge of the ADaM Implementation Guide is assumed in this paper.

DISCLAIMER

All views expressed in this paper are those of the author and are not necessarily those of Roche Products Ltd.

INTRODUCTION

We can all agree that standardisation can lead to many benefits. This is true of internal standards within an individual company and external standards applying to either a collaboration between two or more companies or across the industry as a whole. The gradual introduction and acceptance of the Study Data Tabulation Model (SDTM) in the pharmaceutical industry is a case in point where it is now widely recognised as the standard for data tabulations and is slowly replacing internal standards. Once data tabulations have been standardised the next logical step in any standardisation is the derived datasets used for statistical analysis and work has been ongoing on this for several years within the Clinical Data Interchange Standards Consortium (CDISC) ADaM team. The key principles adopted by ADaM are what would normally be considered good programming practice in the creation of analysis datasets. There are however certain areas within ADaM v2.1 and ADaMIG v1.0 where there is an opportunity for improvement which we could contribute to.

WHY STANDARDISE ANALYSIS DATASETS?

There are many benefits of creating standard analysis datasets; these are applicable from health authority regulators, to the smallest CRO through both big and small pharmaceutical companies. Some of these benefits are equally applicable to internal company standards and a universal standard. For example, benefits relating to the use of standard programs for the creation of both the analysis datasets themselves and for tables and figures can be said to apply to both types of standard.

Two areas where a universal standard provides a benefit over internal company standards is in the provision of analysis datasets to health authority regulators and the sharing of analysis datasets between collaborative partners. For the regulators the key benefit is that they will receive analysis data in a consistent format. So, when moving from a filing from one company to another there is no longer the need to learn a different data structure for essentially the same analysis purpose. This is similar to the benefit for sharing between collaborative partners be it a Pharmaceutical company and a Contract Research Organisation (CRO) or between multiple Pharmaceutical companies working on collaborative projects. In these instances an industry standard avoids the need for a debate between partners over which internal standard to use.

While there are clear benefits of adopting a company standard or more so an industry standard for analysis datasets one point that must never be lost sight of is that there will be instances where a standard does not fit or needs altering to accommodate a particular type of data or analysis. In these instances the need to adopt a non-standard approach should not be questioned as the standard should never be seen to drive what analysis is appropriate or what data is collected. The science and statistics behind these decisions must retain primacy in any decision making process.

THE KEY PRINCIPLES OF ADAM

The key principles of ADaM as stated in ADaMIG v1.0 are:

- Clear and unambiguous communication of the contents of analysis datasets
- Provide a level of traceability back to the input data
- Identify when and how data has been imputed
- Analysis data should be linked to machine readable metadata
- Analysis data should be 'analysis ready'

The high level principles of ADaM are very good and most of these are established in good programming practice. For example dataset specifications, both functional and technical, are key in providing traceability back to the input data, to identify when and how records should be imputed. We can consider our analysis specification to be our metadata as they will in most cases contain information regarding the variable name, label, length, source and the derivation used to create the new variable. Secondly the primary purpose of analysis datasets is to make the programming of outputs easier. So, in that sense, by definition it should be possible to perform statistical analysis from an analysis dataset with minimal programming. I do not mean to detract from the importance of the key principles of ADaM or deride the advantages of an industry standard here but to highlight that none of these key principles could be considered new or revolutionary in the creation of analysis datasets. Where I feel ADaM is at its weakest is where it violates these key principles or does not carry them through to a logical conclusion.

CONSISTENCY ACROSS CDISC STANDARDS

Within the ADaMIG v1.0 there is a section regarding variables which are derived in both SDTM and ADaM. While this section makes valid points regarding possible differences, notably for --TEST/--TESTCD values within SDTM which are a one to many mapping to PARAM/PARAMCD in ADaM. It does not clearly provide definitive guidance other than the ADaM team feel that their model is the one that should have primacy over SDTM for the population indicators and baseline flags.

While as programmers we would normally consider analysis datasets as the appropriate location for both population indicators and baseline flags these are both expected to be provided in SDTM. Consequently it would normally be considered better programming practice to define and derive these variables at the data tabulation level rather than for them to be defined and derived twice, once in data tabulations and a second time in analysis datasets. Even if it real life they are derived within ADaM and then retrospectively joined to the appropriate SDTM domains. While this last approach is discouraged in ADaMIG v1.0 this is more likely to be the practical approach used in many instances due to the involvement of at least Statisticians in the derivation of the populations and in many cases Programmers as well. It is vital that important implementation guidance, such as this, which affects both models, is communicated unambiguously and consistently by both the SDS and ADaM team.

Obviously there are scenarios where ADaM is the only place where some instances of population and baseline variables can be defined. For example if populations are flagged at the individual record level or at the parameter level then it would be appropriate to derive these in analysis datasets, but, for patient level flags these should only be created within SDTM to prevent two different observations being provided to health authorities for what can considered as the same variable. For baseline flagging, it would be preferable to flag baseline once in SDTM. For instances where multiple baselines are required it would be clearer if the baseline used for the primary analysis was flagged in SDTM with additional baselines defined and documented in ADaM as required.

DOES ONE SIZE REALLY FIT ALL?

The ADaMIG v1.0 has been written very much with basic summary statistics and analysis of variance (ANOVA) in mind. For these analyses the structure provided is adequate. Where the ADaMIG v1.0 falls short is in the structure required for event based analysis. For example, while there are limited examples of how data might be structured for time to event analysis within ADaMIG v1.0 it does not give any indication on what each variable should be populated with, for example no advice is provided on the fundamental information as to whether a patient has been censored or not should be stored. Also should the time to event be stored in the analysis day variable (ADY) or in the analysis value variable (AVAL). To leave such important decisions to the sponsor is to not have a standard.

ADaMIG v1.0 is particularly unsuitable for creating listings which contain derived values. Listings traditionally have been some of the easiest outputs to create and validate and ADaM potentially may transform some into the most complicated of outputs. By using ADaM there are examples where these can become very difficult to create from the ADaM structure. Although providing data tabulations to health authorities takes away one need for listings, there will still be other audiences which will need data presented in a listing format.

A different approach would be to have not attempted to fit every type of analysis into a single structure plus a subject level dataset (ADSL) but to have created multiple similar classes of analysis dataset for different types of analysis, for example a class of ADaM datasets could be defined for event based data with clear definitions as to what variables are required, conditional and permissible. Other classes could be defined as appropriate. This approach is analogous to the approach used in SDTM where each domain has inherited characteristics from the domain class to which it belongs.

HOW MUCH TRACEABILTIY IS REQUIRED?

While traceability as a theoretical concept is appealing, the practical application of it can lead to many difficulties when dealing with real life data. The inclusion of additional variables and observations, in all but the simplest cases, inevitably leads to massive duplication and an unnecessary increase in the complexity of analysis datasets. For the simple examples described in the ADaMIG v1.0 where the sequence number (--SEQ) is retained from SDTM in the analysis dataset or where a couple of parameters are used to create a new derived parameter, traceability data is a concept that can be successfully implemented. However, once data is sourced from multiple SDTM domains (or ADaM datasets) the traceability variables become too complex and the amount of duplication increases rapidly. The same is also true of adding extra parameters, for more complicated derivations this could mean adding ten or more additional parameters which will ultimately increase the size of the dataset and the running times of programs. A further complication is there is no method documented for maintaining traceability when new derived parameters are created from other derived parameters, once reaching this stage it is no longer possible to maintain traceability back to the source data.

An improved approach might be to not include traceability data within the analysis data but focus more on ensuring that good quality specifications are provided as part of the metadata. Here, selection criteria for any derivation can be clearly documented in both functional and technical specifications, so as to be clearly explained to multiple audiences. Whereas ADaM does accept there may be instances where including traceability does not aid the clear and unambiguous communication of data, I would argue that this is more often the case than not and it would be better to provide clear guidance on how metadata should be structured to aid traceability than to advise adding complexity to analysis datasets.

ARE THE CRITERIA FLAGS THE BEST WAY FORWARD?

The definition of criteria variables within ADaMIG v1.0 (CRITx and CRITxFL) is an example of the CDISC ADaM team attempting to over standardise the model. In an attempt to standardise additional information about an analysis value the ADaMIG v1.0 have unintentionally violated the key principles of clear and unambiguous communication of the content of the dataset. One of the first things a programmer will do when encountering a dataset for the first time is to examine the variable names and labels of the dataset. By removing the link between the variable name and labels and the contents this removes this possibility.

There are multiple additional problems with this method of assigning variable names. When dealing with multiple analysis datasets across a single study, as is normally the case, unless the criteria variables are made unique across all datasets (thus limiting us to 99 per study due to variable length restrictions) there will be issues when joining multiple analysis datasets together as there is the potential to overwrite records from one dataset with those from another. A second consideration arises when pooling data across a number of studies and potentially across different products. In this instance there would have to be very carefully controlled safeguards in place to ensure that when appending or merging data criteria flags all have the same meaning for the same parameter. This will be especially difficult when no pooling of data is planned and is required quickly thus creating a high risk situation of non-matching data contained within the same variable and the possibility of mistakes.

Another approach would be to state that all information about the analysis value should be stored horizontally using intuitive variable names and labels as opposed to trying to create standard variables for these. The current use of criteria flags as defined in ADaMIG v1.0 is only of benefit for those users who view datasets to examine individual observations. There is only minimal benefit for those who conduct analyses on the data and these are far outweighed by the disadvantages as discussed above.

ANALYSIS FLAGS AND DERIVATION TYPES

A second use within ADaMIG v1.0 of indicator variables is analysis flags (ANLxFL). Analysis flags in many cases will be created in conjunction with the derivation type (DTYPE) variable. As ADaM only creates a new observation for a missing data imputation if required, analysis flags form a necessary part of the model in order to uniquely identify those observations which are required for any given analysis. Using the derivation type alone would not be sufficient to identify records for analysis but may be considered redundant depending on how each analysis flag was defined.

Due to this structure, in many cases the analysis flag will form part of the primary key. As additional records are sometimes required for different analyses the analysis flag, with other identifiers, will be required to uniquely identify observations within the analysis dataset. The inclusion of a primary key is important as it communicates the structure of the data and provides clear documentation for complete joins with other datasets. A further issue around primary keys is that the controlled terminology for analysis flags is “Y” or null. This controlled terminology is not permitted in a primary key. From a programming perspective this is also not good practice as we should always populate variables if we know what they should be. By using the controlled terminology “Y” and “N” we would allow for the use of defensive coding which could highlight the occurrence of null values which obviously have a different meaning than “N”. These could be different data cases not previously considered in which case they may be needed in the analysis. By grouping all these null and "N" together we lose the ability to check for such instances in real life data.

Analysis flags and criteria flags both have similar disadvantages with regard to the naming convention and subsequently working across multiple analysis datasets or studies. Extreme care will need to be taken on each dataset and study within a planned filing to ensure that analysis flags have the same meaning. This is especially the case in all studies which potentially could be pooled. This is because when multiple datasets are either appended or joined together there is not the option of a text variable explaining the analysis flag in the dataset. While all the specifications should be stored in metadata and we should never use the dataset without understanding the specifications it is still a high risk area of ADaM.

CONCLUSION

It is clear that the concept of standardisation of analysis datasets universally is in at least a theoretical sense an unarguable logical next step to take after the adopting of standard data tabulations. The question that remains is, is there a convincing case that ADaM in its current state is a universal standard that can be implemented in practice. I would argue that in its current form ADaM is an immature standard which would require a huge leap of faith on the part of a sponsor to implement. This is particularly difficult to justify in such a high regulated environment where data is so crucial.

Considering the currently available documentation it would be unwise to press ahead with a full scale implementation as there are too many grey areas where sponsors can take opposite decisions based on the same question. While I appreciate the need within any standard for implementation decisions to be made, the current ADaM standard asks more questions than it answers. While the key principles behind ADaM are sound, the implementation of them appears to have only been considered in the simplest of cases, which unfortunately do not appear in many real life studies. Many of the issues discussed will lead to a massive increase in the complexity of the analysis datasets which will violate one of the key principles of ADaM which is the clear and unambiguous communication of results.

As a consequence of all of this I would not recommend any organisation adopting ADaM outside pilot projects as an analysis dataset standard at this time. However, I would strongly encourage all interested parties to contribute to the development of the analysis standard either through joining CDISC teams or by providing comments through the public review. It is through this process that the standard will be improved to a stage where it will be a natural choice to implement, as has already been proven by SDTM.

REFERENCES

ADaM Implementation Guide v1.0 Draft for Public Comment
Analysis Data Model v2.1 Draft for Public Comment
(HTTP://WWW.CDISC.ORG/MODELS/ADAM/V2.1_DRAFT/INDEX.HTML)

ACKNOWLEDGMENTS

I would like to thank

- Olivier Leconte (Roche) for reviewing this paper
- Josephine Gough and Frederik Malfait (both Roche) for providing input

CONTACT INFORMATION

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

Chris Price
Roche Products Ltd.
6 Falcon Way
Shire Park
Welwyn Garden City AL8 7SN
Work Phone: +44 (0)1707 365801
Fax: +44 (0)1707 383145
Email: chris.price.cp1@roche.com