PhUSE 2012

Paper PP06

ADaM in a Pool! A Concept on how to Create Integrated ADaM Datasets

Alexandru Cuza, UCB Biosciences GmbH, Monheim am Rhein, Germany

ABSTRACT

While ADaM becomes more and more the standard to be used for analyses, it is worth trying to establish standard ways of using it.

The range of freedom offered by the ADaM implementation guide is at the same time one of the biggest challenges faced by programmers. Factors such as different study designs, multiple treatments and not least the opportunity to analyze the same data in different ways leads us to the need for a concept on how to create ADaM datasets. While on a study level this can be easily handled, in an integrated analysis it is more complicated.

This paper presents an approach on how to address these challenges and gives a method to solve them in a simple and robust way.

INTRODUCTION

While CDISC specifications for SDTM are - with very few exceptions - well defined and make it possible to start work right away, the ADaM specifications, in order to be able to cover several types of analyses, offer a high degree of flexibility in how we prepare analysis datasets (ADS). Therefore it is obvious that we need a concept on how to construct our analysis database before we can start programming. There are several questions we have to answer in order to have our ADS consistent and compliant:

- How do we want to handle treatments and sub-groups?
- How do we want to put data in place either in a horizontal or vertical structure? etc.

Questions like these are already important in a study level analysis however further ones appear when moving to an integrated analysis, e.g.

- How to integrate studies with different designs?
- How to handle study periods and phases?

We will go through the most important reasons for the concept presented in this paper and specify the goals we would like to reach. Then we will show the solution that was implemented and shortly discuss in the end how this might solve the issues, match the goals as well as any other additional advantages.

SITUATION

ADSL

According to the current ADaM specifications this is the only required ADS. Being a one-record per subject dataset it is not easy to insert all of the needed information.

- ***Treatment handling***

  Pooling several studies together to prepare an integrated analysis is a challenge especially due to different study designs. For example if even one study has switches in treatment within one study period it will not be easy to integrate with the other studies in the pool.

  The implementation guide defines several groups of variables for treatments, e.g. TRTxxP and TRTxxA where xx is the number of the study period. This can lead to an issue if there are different study designs.

  EX1: One or more studies have an open-label period before the blind period while all the others have the opposite design. Then the value of TRT01 in one study is different to TRT01 in another study.

EX2: The study design foresees a switch of treatment within same study period so then we will need e.g. TRT01 and TRT02 which belong to the same study period despite the fact that the numbers tell us that the treatments should be in different study periods.

- ***STUDYID information***
  There are studies which are the follow-up studies of previous ones. Subjects are allowed to continue treatment from e.g. a blinded study into its open-label extension. If an analysis per study is needed, this might be difficult to handle in ADSL because the USUBJID has to be the same in both studies, while we have to keep the STUDYID of each study the subject was enrolled in. Unfortunately the guide doesn't include variables for this. One solution might be to add as many STUDYIDxx variables as needed, but this wouldn't be easy to handle in reporting programs.

- ***Subject-level variables***
  While many subject-level flags or values (e.g. height, weight) could be handled easily and entered into ADSL, there are often others (.e.g sub-groups) which need to be derived in other ADS but they cannot be entered in ADSL (as recommended by the ADaM guide) due to potential circular dependencies between the ADS and ADSL.

OTHER

- The need for different kinds of analyses on the same integrated data.

- Updating the pooled database with new data from previous integrated studies. This might be easy as there might not be a need for structural changes to ADS.

- Updating the database with new studies. This will be a challenge if the new studies have completely different designs (please refer to ADSL topic above)

GOALS

Based on the situation above we have identified some objectives we would like to achieve.

- **Treatment handling:** This has to be handled better than described in the ADaM implementation guide. It is obvious that in its current form ADSL does not match the needs for integrated analyses.
  Even if it is possible to put all treatments in ADSL without creating confusion (see TRTxx issue) this will still lead to an increased amount of code in TFL programs.
- **STUDYID:** A method to keep this information has to be implemented.
- **Subject-level information:** Values, flags, sub-groups which are one per subject, but dependent on data in other ADS (e.g. due to derivation rules applied) need an efficient way and place to create them in ADS. Circular-dependencies must be avoided.
- **Different analyses on same data pool:** In order to make this possible the ADS structure has to achieve some grade of independence from the Statistical Analysis Plan (SAP).

SOLUTION

Treatment handling is the most important point of this concept. The main idea behind it is to take, as far as possible, everything that is dependent on treatment out of the ADS creation. Almost all ADS will be programmed without treatment dependent variables as these will be added at the end by a separate program.
Below are the steps in the order that we implemented them.

ADTRTG

This is the so called generic treatment dataset. "Generic" because it is independent of any SAP rules. Based on SDTM EX panel it contains one record per STUDYID, USUBJID and continuous treatment. "Continuous" means the same study medication, dose, dose unit, formulation, frequency etc. Furthermore each record contains the start and end dates of the continuous treatment. If we have to consider drug interruption then a new record (=interval) for the same treatment has to be created.

ADTRT
Here we apply the SAP rules and treatment definitions using *ADTRTG*. For example some treatments might be combined and others might be ignored in the actual analysis. The structure is the same as in *ADTRTG*.
In order to make code re-usable and more robust we implement a so called treatment matrix. This is a file containing the mapping of existing treatments to the ones described in the SAP. If we need to perform a new analysis on the same integrated data we only need to replace this file with a new one.

ADSL
Due to the issues identified above we apply a "diet" to ADSL by keeping only the required variables as defined by the ADaM implementation guide (e.g. population flags).

All other ADS
The rest of ADS can be created at this stage, but without including any treatment dependent code, e.g. treatment or period assignment to an Adverse Event (AE). Exceptions are, for example, date imputation rules or time since first study medication dose, for which we use *ADTRT.*

ADSBG
This is an ADaM Basic Data Structure (BDS) dataset containing all subject-level flags, sub-groups etc. in a vertical structure. By having this structure we resolve the circular dependency issue mentioned above and at the same time ease the addition of new sub-groups and simplify the programming of reporting items.

PR_PH_RD.SAS
This program assigns a study event (AE, medication, measurements over time like vitals signs, laboratory etc.) to a treatment, a study period (**PR**), a study or treatment phase (**PH**) and calculates relative days (**RD**).
This also uses *ADTRT*. For example to assign a study event to a treatment, the program looks up in which treatment interval (=*ADTRT* record) the event occurred.

CONCLUSION

- **ADSL** issues were solved by reducing this panel to the minimum number of variables required.
- **STUDYID:** By keeping this variable in ADTRTG and ADTRT and in all other ADS as a copy of SDTM STUDYID, reporting by study is possible.
- **Subject level information:** Moving this into ADSBG avoids circular dependencies between ADS and ADSL and also simplifies report programming.
- **Different analyses on the same integrated data:** By replacing the treatment matrix used in ADTRT and re-running PR_PH_RD.SAS this is easily implemented.
- **Additional advantages:**
  - High level re-usability of programming code not only when reporting on the same integrated data, but also when moving to another project.
  - Although the concept was mainly created for integrated analyses, it can also be applied to a single study analysis.

At the very end we should also mention that although the concept was implemented successfully in three integrated analyses, at the time of writing this paper the ADS database was not yet submitted. Therefore as yet we do not have any feedback from the regulatory authorities.

CONTACT INFORMATION

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

Alexandru Cuza
UCB Biosciences GmbH
Alfred-Nobel-Str. 10
40789 Monheim, Germany
Work Phone: +49 2173 48 1191
Email: alexandru.cuza@ucb.com
Web: www.ucb.com