PhUSE Paper 2016

Automating ADaM - How to Efficiently Create CDISC ADaM Specifications and Automate their Transformation into Datasets

Abstract

In December 2014 the FDA published binding guidance that will require study data to be submitted in electronic format in conformance to CDISC standards. An important component is the production of the analysis datasets following ADaM, whose creation can present challenges due to their size, traceability requirements and the need for consistency between them. The use of a central metadata repository (MDR) to store data standards, along with the application of analysis concepts to underpin variable derivations and, finally, the use of generic code to support the analysis concepts wherever they are applied can have major benefits in addressing some of these challenges. The aim of the poster is to highlight issues in producing ADaM datasets and to suggest a centralized, generic and automated approach.

Challenges

- ADaM datasets are large and repetitive in structure, therefore tedious to program manually.
- The task of managing consistency between common variables across datasets can be onerous.
- The ADaM standard requires metadata showing clear and full traceability, which can be complex to handle. Questions arise over how to store it, how to present it and make it clear to a reviewer, how to make it accessible for tools and people?

Solutions employed at Roche

Automated programming

In order to create the ADaM datasets efficiently Roche has developed a series of standard programming modules in the form of standard SAS macros.

For the Basic Data Structure (BDS) datasets, the analysis parameters are stored in files in machine readable format and are processed automatically by the macros.

The programming modules can be used independently of each other and individual modules can be replaced with custom code, allowing users the flexibility to deviate from the standard if required. For example, if a user wanted to employ a new method to calculate patient age, then the user would be able to provide a new module within the defined structure without affecting the overall program. Pre- and post- conditions for the application of modules are set out for the user.

Development of analysis concepts

Analysis concepts established largely by the biostatistics group have been put in place to drive and underpin the variable derivations and are applied consistently across parameters or to variables across datasets where appropriate.

An example is the analysis concept supporting “PCHG”, Percentage Change from Baseline, which is common across many BDS datasets. The analysis concept is as follows:

*"(Change from baseline value/Corresponding baseline value) * 100*
Baseline value must be non-zero."

The corresponding associated variable derivation in each relevant BDS dataset is set in the specification for each dataset. For example in ADEG (ECG Analysis) the corresponding derivation is:

"Set to (Change from Baseline [ADEG.CHG] divided by Baseline Value [ADEG.BASE]) multiplied by 100
Do not compute if Baseline Value [ADEG.BASE] is 0."

The concept is applied in all ADaM datasets where percentage change from baseline is calculated.

A benefit of applying analysis concepts is that they reduce the risk of discrepancy between specifications and code. A concept usually leads to a generic macro being called within a standard program.

GDSR

What is it the GDSR?

The Global Data Standards Repository (GDSR), the metadata repository at Roche, is the central source for standard study metadata. It is a triple store based on semantic web technology, describing and linking data using Resource Description Framework (RDF) language.

What are the functions of the GDSR?

Our ADaM specifications are generated directly –from the GDSR directly from a central "single source of truth".

All the dataset, variable and value level metadata for our ADaM datasets are populated in the GDSR from the point of initial creation, offering consistency, accuracy and traceability. Each analysis concept or piece of metadata only needs to be defined in the GDSR once, and will be displayed in the GDSR generated specs wherever it is referenced.

The specifications can be produced using a combination of a subset of standard defined and study specific implementation.

Advantages of approach

The use of the GDSR facilitates the traceability of analysis data. An example of the complexity of presenting traceability is the variable ADEG.AGE, whose immediate predecessor is displayed in the specification for ADEG as "ADSL.AGE", but whose attributes are set at the "source" DM.AGE. The presentation of the immediate predecessor of this variable (ADSL.AGE) as well as its attributes is enabled not by copying or reproducing the information, but by storing it once and only referencing it where needed from the source (in this case DM domain).

Equally, because the specifications and concepts are retrieved from the same source, the same programming modules can be identified and linked. This extends to the creation of the standard programs, so that wherever the same analysis concepts and derivations are used, the same generic programming modules (eg. SAS macros) are applied.

For example, wherever an Analysis Date Imputation Flag is employed in an ADaM dataset, the following generic derivation is applied:

> *"*DTF variables represent the level of imputation of the *DT variable based on the source SDTM DTC variable.*
> **DTF = Y if the entire date is imputed.*
> **DTF = M if month and day are imputed.*
> **DTF = D if only day is imputed.*
> **DTF = null if no imputation performed.*
> *If a date was imputed, *DTF must be populated and is required."*

This algorithm is in turn is supported by a generic SAS macro.

The use of machine readable analysis parameters also ensures consistency and means that tools can be used to perform some of the quality checks, e.g comparing the derived variables with the specifications.

This centralized, generic and automated approach minimizes the risk of discrepancy across datasets and facilitates their production.

Future direction

The ultimate goal is full traceability, which we aim to achieve by eventually entering study level metadata into GDSR. Once study level metadata is entered then this will enable us to choose options within GDSR, which will in turn automatically create tailored programs and macros calls. A first step would be to store in the GDSR each instance of the implementation of the standard, ie. the metadata from each study.