Paper DS03

Qualitative ADaM datasets and define.xml without using ADaM specs

Pieter Coppens, SGS Pharma - Clinical Research, Mechelen, Belgium

ABSTRACT

Both the structure and derivation rules for Analysis Data Model (ADaM) datasets are typically defined in ADaM specifications before statistical programming begins. However, at such an early stage, key documents are often still under development and clinical data are still incomplete. Moreover, the data are not fully cleaned yet. As a result, assumptions must be made which introduces the risk of incomplete or inaccurate ADaM specifications, in turn leading to mistakes in ADaM datasets.

In this paper we present an alternative approach that involves delivering a testrun analysis without the need for predefined ADaM specifications. This method uses actual study data and is guided by a central ADaM model. Statistical programmers develop the datasets based on a stable version of the statistical analysis plan (SAP), while an independent validator performs derivations in parallel to ensure unbiased results. Any discrepancies are resolved through discussion and refinement.

In parallel with the testrun, submission-ready metadata are generated in Define.xml format. Dataset structure and codelists are extracted directly from the datasets, while study-specific derivations are added by the developer.

By integrating dataset development, validation, and metadata generation into a single workflow, this approach supports the creation of high-quality ADaM datasets and define.xml early in the process. At the same time, it ensures compatibility with the latest standards while still adapting to evolving study requirements.

INTRODUCTION

ADaM datasets are datasets that serve as a backbone to generate the tables, listings, and figures (TLFs) used in the study report and other statistical analyses such as independent data monitoring committee (IDMC) activities.

ADaM specifications (specs) are a study-specific data model that defines how these datasets are derived from SDTM datasets. These specs contain details about the dataset structure (i.e. list of required datasets and variables), content (i.e. values to be used), and source/derivation logic for each variable and parameter.

Typically, ADaM specs are created as a large spreadsheet with several tabs that includes the following aspects:

- Domain list: A comprehensive list of all ADaM datasets expected to be created for the study, together with their properties (such as structure, keys, class, and label)
- Dataset-specific sheets: One sheet for each dataset listing all variables required for analysis and their properties. This includes codelists of allowed values, source variables or derivation algorithms, variable lengths and labels, ...
- Codelist sheet: A list of all potential values applicable for the concerned variables. These values can be specific for the study (e.g., age group) or originate from published controlled terminology (e.g. DTYPE).

ADAM SPECS PROCESS

The process with ADaM specs starts with creating a stable Statistical Analysis Plan (SAP). This document describes in detail which analyses are needed to meet the study objectives defined in the protocol. A version of the SAP is drafted by the CRO and subsequently reviewed by the sponsor to eventually reach a stable SAP version.

Next, study-specific ADaM specs are created based on this stable version of the SAP and other available documents, such as mock-ups of the TLFs, the (annotated) case report form (CRF), and the expected SDTM structure. Once developed, the specs are reviewed by the sponsor and any comments are implemented/resolved prior to the start of programming activities.

Programming starts with the developer (primary programmer) developing a testrun analysis, i.e. ADaM datasets and TLFs, mainly based on ADaM specs as well such as the stable SAP, mock TLFs, CRF, protocol, etc.. The testrun analysis is validated by a validator (secondary, independent, programmer) based on the same information (e.g. by double programming). After the developer has implemented all validator comments, the testrun analysis is released to the sponsor for review. Often, the ADaM specs are also updated based on issues encountered during this first programming stage, such as unexpected SDTM structure, changes to the SAP, etc.

After study completion and database lock, the final analysis is run by the developer, taking any sponsor comments into account. The analysis is then validated by the validator and finally released to the sponsor. The corresponding metadata (in Define.xml format) are created after the final analysis is ready, using the ADaM specs as input for their creation. The final step in the analysis process is the release of the define.xml to the sponsor. The entire process is shown in Figure 1 below.

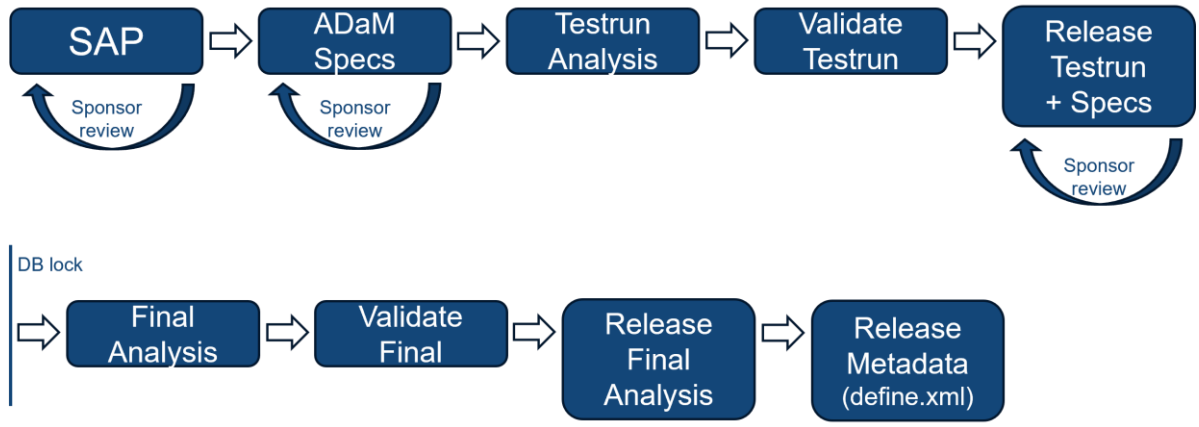

Figure 1: Process with ADaM specs.

Using this process with predefined ADaM specs poses several challenges:

1. The initial creation of specs is time consuming. This is mainly because the ADaM dataset design is based on what is essentially a theoretical approach. It takes time to understand the structure of the SDTM data and to specify all necessary parameters when (actual) study data are not yet available.

2. Since both developer and validator start from the same derivations and algorithms described in the ADaM specs, there is no independent validation of these aspects based on actual study data. This can lead to confirmation bias from the validator (and developed) by over-reliance on the derivations as described in the ADaM specs.

3. During programming, the ADaM specs need to be actively maintained. The structure of the SDTM domains at database lock often differs from the original SDTM design on which the ADaM specs were based. In addition, the SAP can be updated in case new analysis needs arise. These changes necessitate continuous updates and revisions to the ADaM specs, not to mention situations when derivation algorithms predefined in the specs are not sufficient to cover all actual study data.

4. ADaM specs are unique for each individual study. Not only are they of course tied to the design and requirements of their corresponding study, but they are also made based on a specific version of the ADaM standard, implementation guide, and controlled terminology. In case of version updates, the specs cannot be easily reused without significant adaptations for a different study, even if the design and analysis requirements are very similar.

PROCESS WITHOUT ADAM SPECS

To overcome these challenges and increase programming efficiency, a process was designed that does not require creating ADaM specs upfront.

The first step of this process is the same: development of a stable SAP through draft and sponsor review cycles.

However, the next step deviates from the approach described earlier: ADaM specs are no longer developed. Instead, development of the testrun analysis starts immediately based on the available documentation and SDTM data at that time. In the absence of ADaM specs, programming uses a central ADaM model.

The central ADaM model is a library containing fundamental metadata of all allowable ADaM domains, variables, and controlled terminology. In that sense, the model is a translation of the CDISC ADaM implementation guides (ADaM IGs) as per SGS interpretation. Since this model is key to the process, it is developed and maintained on a continuous basis by dedicated standards experts. They keep the central ADaM model up to date with the latest industry and CDISC standards as well as specific sponsor standards. The ADaM datasets are thus programmed based on this central model and further finetuned to the study-specific needs, during testrun development.

To accommodate for the fact the sponsor does not receive ADaM specs, draft ADaMs (ADaMs that have not gone through the full validation cycle) are provided to the sponsor at an early development stage, prior to testrun delivery. These draft ADaMs give the sponsor an idea of the structure of the ADaM datasets, so they can verify it and start preparing for review at their end.

The validation process is similar to the process with ADaM specs, except for the absence of these ADaM specs of course. The derivations (a crucial part in ADaM development) are now developed fully independently by both developer and validator. This is a deliberate choice to avoid the validator being biased. This approach mitigates

the risk of errors in derivation algorithms. In case of differences in the results obtained by developer and validator, the root cause is discussed by both together. They then discuss the best approach and communicate it to the sponsor. If necessary, the SAP can be extended.

After validation, the testrun analysis is released to the sponsor. Where in the process described earlier this package would also contain ADaM specs, the full define.xml is released instead in this process. These detailed metadata are created by the developer parallel to working on the testrun analysis. The central model is also used to generate a substantial part of the define.xml automatically. In addition, the created ADaM datasets are used to automate other parts of the define.xml, such as creating code lists. Study-specific aspects (especially the derivation algorithms) still need to be added manually by the developer. A review cycle of the testrun analysis by the sponsor is again foreseen.

After study completion and database lock, the final analysis is run by the developer, then validated by the validator, and finally released to the sponsor. This time, instead of creating the define.xml *after* the release of the final analysis, it is available already from the first testrun and is thus provided together with the release of the final analysis (see Figure 2 below).

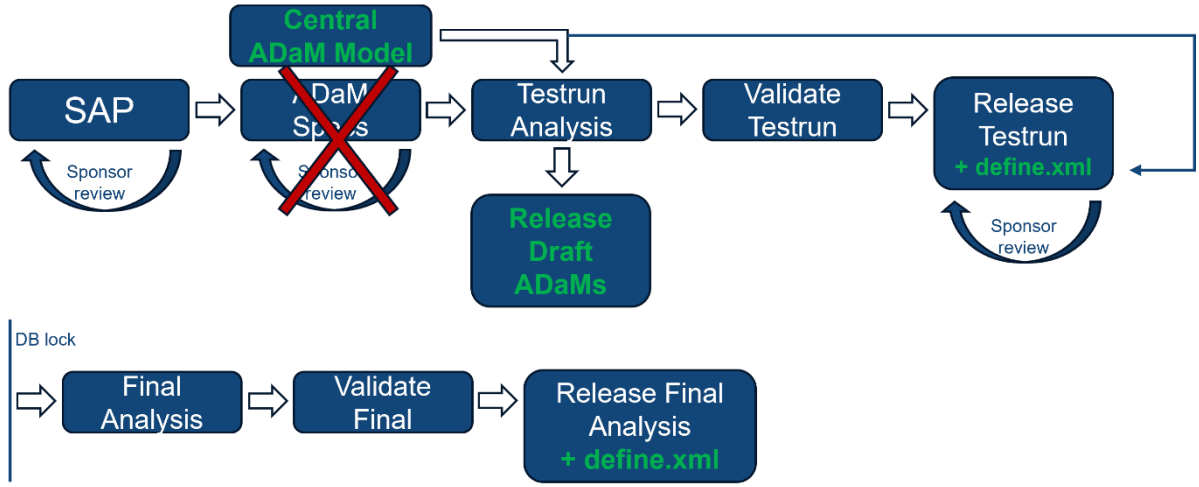

Figure 2: Process without ADaM specs.

So, what are the advantages of not relying on predefined ADaM Specs?

1. Quality of the ADaM datasets:
    - Programming starts immediately based on actual data, rather than starting from a theoretical design. This means that it is easier for both programmers to gain an intimate understanding of the data and detect special cases that may be overlooked otherwise.
    - As mentioned before, the fact that derivation algorithms must be developed independently by the developer and validator avoids bias and assures quality.
    - Since the ADaM development starts from the central model directly, automated checks are built into the programming process to verify whether the final product is compliant with the latest ADaM model. This increases consistency and consequently, quality, across studies.

2. Increased efficiency:
    - The effort of creating ADaM metadata and their maintenance throughout the analysis development cycle has been reduced significantly.
    - Programming starts sooner, which translates to shorter timelines.
    - The define.xml is available sooner (already at testrun) and substantial parts can be generated automatically from the central model in combination with the analysis datasets.

3. It is more interesting for the programming team:
    - Programmers are highly trained and do not only focus on programming prespecified algorithms. They are actively involved in the translation of all source documentation into a correct ADaM structure.
    - Working on actual study data makes the work more tangible and interesting than first creating abstract, theoretical specifications that will change anyway.
    - The developers and validators have open discussions to arrive at the best solution together if their result does not match, rather than trying to replicate what was specified in a predefined set

of ADaM specs.

4. The sponsor has access to more accurate and complete (meta)data at an early stage:
    - Since draft analysis datasets are provided early in the process, the sponsor can already start preparing review at their end earlier.
    - This review will be on a testrun package that contains everything necessary, including a fully developed define.xml with study-specific derivation algorithms. The define.xml contains more information than the ADaM specs alone.

CONCLUSION

The traditional process of using predefined, study-specific ADaM specs poses challenges, especially for complex studies. This paper presents an alternative approach without ADaM specs, offering greater quality, flexibility, and efficiency. This process evolved through years of hands-on experience and has been well received by sponsors.

ACKNOWLEDGEMENTS

Many thanks to Pieter Lemkens for his help in writing this article and to Jeroen Den Herder, Wouter Milis, and Rukun Hinz for their help creating the abstract, presentation, and this paper.

CONTACT INFORMATION

Pieter Coppens, PhD
SGS Pharma - Clinical Research
Generaal De Wittelaan 19A bus 5
BE - 2800 Mechelen

EUROPE: +32 15 27 32 45
AMERICAS: + 1 877 677 2667
Email: clinicalresearch@sgs.com
Web: www.sgs.com/cro