

Please Take Care of your Metadata

Angelo Tinazzi, Cytel Inc., Geneva, Switzerland

ABSTRACT

Most of the companies validate metadata (define-xml) using a standard set of rules defined by CDISC and available in the "de-facto" standard tools such as Pinnacle21; the validation is usually performed on individual packages, SDTM or ADaM. However, these set of rules, and the way are implemented, lack of traceability checks from ADaM to SDTM.

With this paper, I would like to stress on the importance of metadata and data consistency between SDTM and ADaM, between ADaM datasets and how at Cytel we were able to fill the gap with a number of internal SAS standard programs and macros.

For example, if an ADaM variable has *origin=predecessor* the variable exist in SDTM and with same attributes, or if a variable from ADSL is replicated into other ADaMs the attributes and content is not changed. Moreover, the tools re-use SDTM metadata when a variable in ADaM define has *origin=Predecessor*.

INTRODUCTION

define-xml for SDTM and define-xml for ADaM are usually treated by sponsors and vendors as an independent and separate deliverable; the same is true for the Study Data Reviewer Guide for SDTM (cSDRG) and Analysis Data Reviewer Guide for ADaM (ADRG). However, there are many information between SDTM and ADaM that are repeated and so for their documentation. This generates very often inconsistencies between for example the same information reported in SDTM define-xml and ADaM define-xml. Some of this sort of 'redundant' information, due to the type of file, for example PDF in the case of the cSDRG and ADRG, needs to be checked visually i.e. the version of the MedDRA dictionary reported in the "Study Data Standards and Dictionary Inventory" should be same in cSDRG and ADRG.

For define-xml instead, given the fact it is machine readable, we can make 'consistency' checks programmatically.

WHAT MAKES A DEFINE-XML CONFORMANT?

In order to be conformant, a define-xml needs to be a valid according to some general XML principles and of course conformant according to the define-xml standard¹:

- XML code must be syntactically correct
- Valid according to the associated XML Schema
- Metadata vs dataset attributes and content
- Appropriateness of comments and methods

For example, you can check if define-xml is conformant to define-xml standards using tools like Pinnacle21. Two round of validation are required with this tool:

- Validate define-xml alone → Make sure your define-xml is compliant with CDISC define-xml standard
- Validate datasets with define-xml → Make sure what you have planned (define-xml) is what you have done (datasets)
 - Any missing variable in define-xml?
 - Any missing variable in the datasets?
 - Variables attributes match define-xml attributes?
 - Any occurred term that is not described in the define-xml?

¹ <https://www.cdisc.org/standards/data-exchange/define-xml>

SDTM METADATA VS ADaM METADATA

Unless you have already in place a good metadata repository, eventually with some linked data approach, these validations do not make any major check of data nor metadata provided in SDTM define.xml vs metadata provided in ADaM define.xml. The only exception with Pinnacle21 is the possibility of including in the validation of ADaM datasets, the SDTM datasets DM, EX and AE so that some basic checks can be performed between SDTM and ADaM datasets content. For example checking that if USUBJID is present in ADSL this should be present in DM, or that all Adverse Events reported in AE are also in ADAE, etc.

There are however no other checks to verify for example variables that have been copied from SDTM to ADaM (origin=predecessor) exist in SDTM i.e. they are not misspelled in the ADaM define 'predecessor' metadata, and all attributes in SDTM are same in ADaM, for example the label, the type, the code-list, etc.

There is in few words a risk of lacking metadata but also data traceability.

THE PROBLEM WITH DATA AND METADATA TRACEABILITY

As shown in Figure 1 there are three type of data in the SDTM/ADaM 'Ocean':

- SDTM Only: source variables and records not used in ADaM i.e. Screen Failures, IE domain, Suppl. Lab. Data, etc.
- SDTM and ADaM: variables and records copied from SDTM to ADaM → Traceability Issue 1
- ADaM Only: derived or assigned variables / new records in ADaM for analysis purpose → Traceability Issue 2

Traceability issue 1, can be easily programmatically checked even without a complex metadata "system":

- Check for ADaM Origin=Predecessor
 - o Keep all variables attributes from SDTM
 - o Bring code-list defined in SDTM
 - o Content must be not changed

The same applies when Predecessor is another ADaM dataset.

Traceability issue 2, requires more Independent and Manual Review

- Check for Clear Pattern of Derivations
- Creation of Records if needed i.e. impute missing observations
- Clear description of derivations



Figure 1: SDTM and ADaM datasets

A GOOD INTERIM SOLUTION

As mentioned earlier, good future tools will probably resolve this problem. I'm thinking about for example the idea beyond the CDISC 360¹ project where, by properly setting up of linked metadata, you could reduce the "silos" and therefore treating a study from start-to-end, where study information, either protocol characteristics or form design or controlled terminology or statistical analysis requirements, are all linked together.

However, we are either not yet there or in many situations sponsors are still treating CDISC deliverables, either SDTM or ADaM, and their submission-ready deliverables i.e. define.xml and reviewer guide, as a sort of post processing process where CDISC metadata are defined retrospectively.

¹ <https://www.cdisc.org/cdisc-360>

PHUSE EU CONNECT 2020

In such a situation, we need to handle differently with some ad-hoc processes, very often one for SDTM and one for ADaM, especially in the situations where SDTM and ADaM, and TFLs is created by different vendors or groups.

CAPTURING DEFINE-XML METADATA WITH SAS

Assume you are responsible for the analysis part only, so you are responsible for creating ADaM package, while SDTM is created by a second vendor or group. You then receive the completed SDTM packages for you to use as source for your analysis.

Because metadata for creating define-xml are not defined by you or by your organization, you really want wanted handle electronically your ADaM metadata and you want to reduce the burden of metadata creation especially if for variables coming from SDTM you already have available metadata in the SDTM define-xml.

There might be several ways of capturing or handling metadata from an existing define-xml, but with SAS you can capture almost all define-xml metadata with the *xmlv2* LIBNAE engine. For example, consider the following code:

```
filename define      "..\define.xml";
filename map         "..\Work\defineXMLautomap.map";
libname define      xmlv2 automap=replace xmlmap=MAP;
libname defineMt     "..\Work";
proc copy inlib=define outlib=defineMt;
run;
libname define clear;
```

With these statements, you can capture into a set of datasets most of the metadata available in define-xml. Figure 2 shows the datasets automatically created.

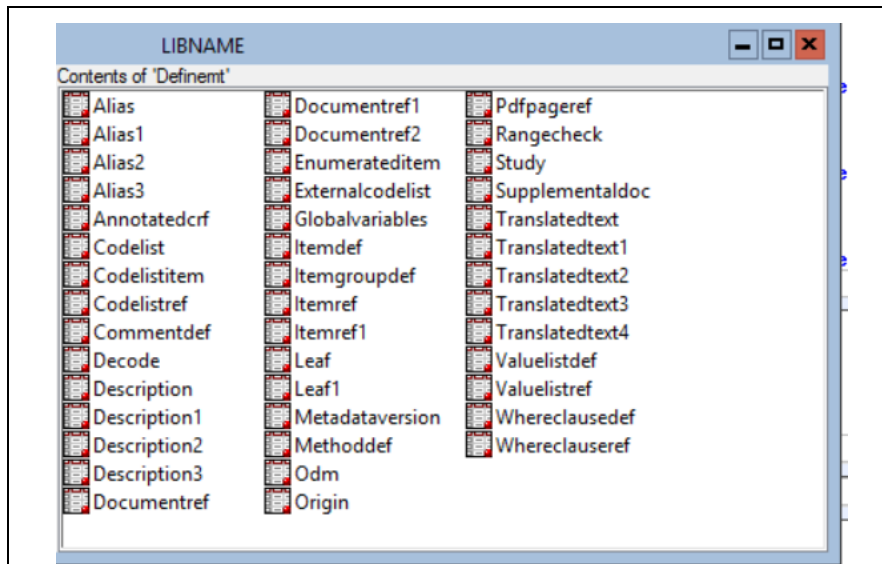


Figure 2: SDTM and ADaM datasets

For example figure 3 is an extract of dataset *ItemGroupDef* created from the define-xml “import” containing the dataset metadata.

ItemGroupDef_Name	ItemGroupDef_Repeatin	ItemGroupDef_IsReferenceData	ItemGroupDef_SASDat	ItemGroupDef_Purpose	ItemGroupDef_Structure	ItemGroupDef_Class
TA	No	Yes	TA	Tabulation	One record per planned element per arm	TRIAL DESIGN
TE	No	Yes	TE	Tabulation	One record per planned element	TRIAL DESIGN
TI	No	Yes	TI	Tabulation	One record per I/E criterion	TRIAL DESIGN
TS	No	Yes	TS	Tabulation	One record per trial summary parameter value	TRIAL DESIGN
TV	No	Yes	TV	Tabulation	One record per planned visit per arm	TRIAL DESIGN
CO	Yes	No	CO	Tabulation	One record per comment per subject	SPECIAL PURPOSE
DM	No	No	DM	Tabulation	One record per subject	SPECIAL PURPOSE
SE	Yes	No	SE	Tabulation	One record per actual element per subject	SPECIAL PURPOSE

Figure 3: Example of datasets metadata captured from define-xml (dataset ItemGroupDef)

PHUSE EU CONNECT 2020

If you want to capture all metadata, you would need to define a mapping file where you provide a more complete set of mapping rules so that the xmlv2 engine can correctly interpret the content of your define-xml [1].

By being able to have your metadata represented as a set of datasets, you can program some checks to verify that your ADaM metadata are compliant with SDTM metadata, or you can “push” SDTM metadata to ADaM metadata i.e. when a variable is copied from SDTM to ADaM.

CHECKING ADAM METADATA VS SDTM METADATA

In the example in figure 3, we were able to detect a possible mistake in the ADaM metadata. The two variables, AEHLTGCD and AECAT, do not exist in the SDTM while the ADaM metadata are indicating Origin=Predecessor.

Issues	Dataset	Variable	Label	Codelis	Origin	Predecessor
[ERROR: Variable does not exist in predecessor dataset]	ADAE	AEHLTGCD	Group	MEDORANUM	Predecessor	AE.AEHLTGCD
[ERROR: Variable does not exist in predecessor dataset]						
[ERROR: A codelist is referenced but the codelist	ADAE	AECAT	Category	AECAT	Predecessor	AE.AECAT

Figure 4: Example of automatic SDTM/ADaM metadata check

ADAM METADATA INHERITATING SDTM DEFINE-XML METADATA

Because many of the information available in the ADaM datasets, records and variables, are coming from SDTM, it will be good to have the same metadata and therefore the same information in ADaM define-xml when for example a variable is copied from SDTM. Figure 4 is an example where the variable MHENRTPT copied into the ADaM dataset ADMH from the MH SDTM dataset. In SDTM define-xml, the variable was having a code-list. However, the author of the ADaM define-xml did not assign any code-list for the same variable in the ADaM define-xml; this was detected by the tool and the code-list was assigned to the same variable in ADaM and the code-list information also copied in the controlled terminology section of the define-xml.

Issues	Dataset	Variable	Label	Codelis	Origin	Predecessor
[NOTE: CT assigned from SDTM define]	ADMH	MHENRTPT	End Relative to Reference Time Point	ENRTPT	Predecessor	MH.MHENRTPT

Figure 5: Example of Automatic check and Inheritance of Metadata from SDTM to ADaM

CONCLUSIONS

define-xml and reviewer guide(s) are two important and strategic elements for the success of your data submission. Their quality is your “business card” to the agency. One of the key is the “consistency” of the information reported between SDTM and ADaM package. Discrepancies between the two might be spotted by the reviewer, therefore generating questions or, if not spotted, could be the source of misunderstanding.

REFERENCES

1. “Accessing the Metadata from define XML “, L. Jansen, PharmaSUG 2018

RECOMMENDED READING

1. “How to ensure quality in data submission“, A. Tinazzi, PharmaSUG-China 2019

The Good Data Submission Doctor - New ADaM Implementation Guidance – Cytel Blog

<https://www.cytel.com/blog/new-adam-implementation-guidance>

PHUSE EU CONNECT 2020

CONTACT INFORMATION

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

Angelo Tinazzi

Cytel Inc.

Route de Prè-Bois 20

1215 Geneva – Switzerland

Email: angelo.tinazzi@cytel.com

Web: www.cytel.com

Brand and product names are trademarks of their respective companies.