PharmaSUG 2016 - Paper AD18

Consider Define.xml Generation during Development of CDISC Dataset Mapping Specifications

Vara Prasad Sakampally, Vita Data Sciences, Waltham, MA
Bhavin Busa, Vita Data Sciences, Waltham, MA

ABSTRACT

As per the submission data standards set by the FDA for the new drug application, the sponsor has to provide a complete and informative define.xml as part of the dataset submission packet along with other required components. FDA specifies to submit the metadata information of a submitted dataset as per the CDISC Define-XML file standard because of its virtue of both machine and human readable properties. Most sponsors consider generating or receiving define document from their vendor during the final steps of dataset submission. There are multiple papers discussing different approaches to create the define.xml file. In this paper, we are presenting an approach where we have used Pinnacle21® validator to generate define.xml during the specification and dataset development phase. This paper also provides an insight on using Pinnacle21 validator as a tool with a constructive approach to generate define.xml and validation of the datasets during the SDTM development lifecycle.

INTRODUCTION

The standardized clinical study datasets will be required in submissions for clinical and non-clinical studies that start on or after December 17, 2016 [1]. As noted, it will be expected that all the trials conducted after that date must use study data standards that are listed in the FDA Data Standards Catalog (DSC). This means that all studies going forward must utilize CDISC SDTM and ADaM standards for their tabulation and analysis datasets respectively and should consist of data definition file to describe the metadata of the submitted electronic datasets.

As stated in the FDA Study Data Technical Conformance Guide that the data definition file (define.xml), “is considered arguably the most important part of the electronic dataset submission for regulatory review”. A well-defined, organized and standardized define.xml minimizes the time needed to familiarize with the metadata information of the submitted datasets which significantly decreases the overall time for review process. Also one should note that an insufficiently documented define.xml is a common deficiency that reviewers have noted. And so generating a properly

The figure 1 below demonstrate typical SDTM development life cycle. The number annotated on each box represent the process and sequence in which they are developed and implemented. Upon development and QC of the SDTM datasets, a programmer passes the datasets through Pinnacle21 validator to check for the compliance. In cases where the study specific define.xml is available, it is also passed through the validator along with the SDTM datasets. Although the typical process employed in the industry is to generate define.xml at the final submission stage and assumes that it has to be done after the SDTM/ADaM development lifecycle.

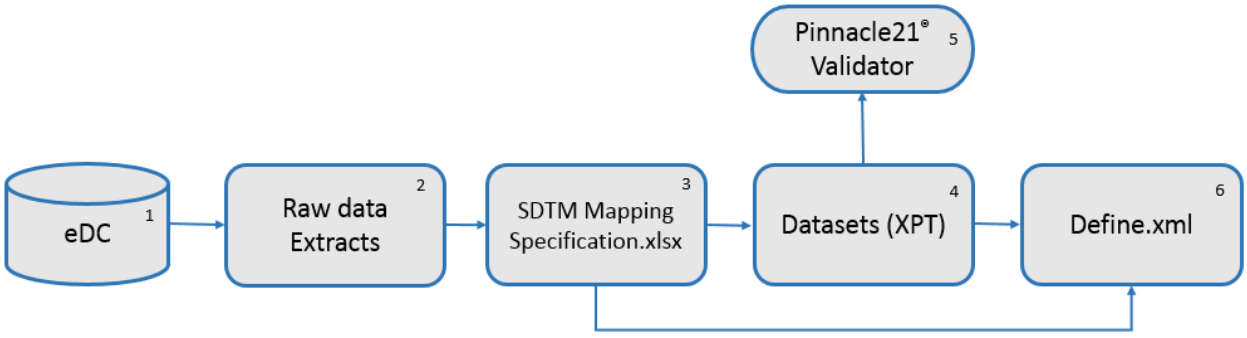

Figure 1. Typical process followed in industry

Generating the define at the end of the study cycle has couple of advantages like, if any of the anticipated domain is not submitted for the reason of not having any records, that can be omitted and the time required to incorporate the details of that domain into define.xml can be saved. A complete list of the controlled terminology can be obtained from the actual data and can present only the values from data.

The major disadvantages of generating define at the end of the study are as below:

- If any change in the metadata information is required by realizing from the Pinnacle21 report then the entire validation cycle has to be performed.
- If there is any information that is missed on define then there is a high chance that it might not surface during the review cycle and it being generated at the end might not be subjected to a reasonable review cycles.

We propose that the define generation should be done during the SDTM/ADaM development lifecycle, specifically at the time of writing dataset mapping specification. This paper walks you through the general procedure that can be used to make the process of define generation at the early stages of the dataset generation process.

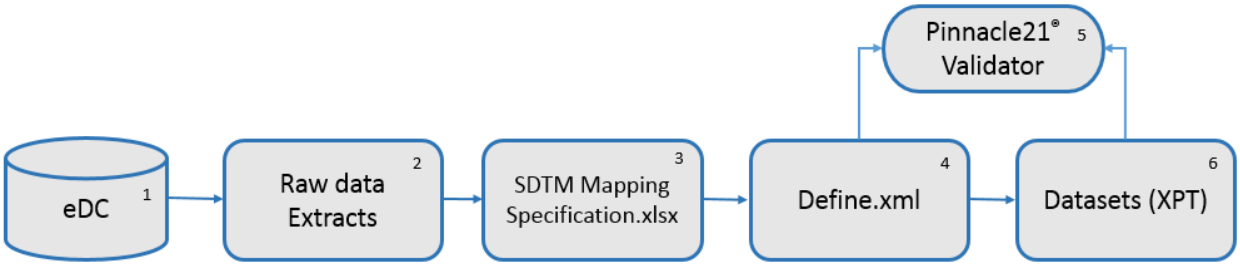

Figure 2. Our proposal is to generate define.xml before creation of datasets

Generating define at the start of the dataset development lifecycle has below advantages:

- Generating define upfront will subject it to multiple review cycles. An example of such a case is that it can be used by the programmers during coding and to navigate from one data point to other using the hyperlinks which helps to identify of any broken links without any additional effort.
- Helps in speeding the review process by the client as all the information will be on single file which is hyperlinked for easy navigation.
- Helps to check for the metadata differences in comparison to standard at the very beginning of the study that helps to makes changes without much effort.
- A SAS® based macro can be used to annotate the CRF right at the start of the study and any changes that has to be done during the course of the study can be made. Annotating CRF by relying on the define helps to maintain consistency in annotation with the Define.
- A SAS macro can be used to generate the dataset shells with the attributes and can pass through the Pinnacle21 validator along with define to identify any missing variables that are required or expected. The variable order in the dataset can also be validated with this approach.

We at Vita Data Sciences used an approach to use a specification template which mimics the file that Pinnacle21 uses as input to generate the define.xml. The components that are needed for define 2.0 and the corresponding pieces of the datasheets required by Pinnacle21 are as below.

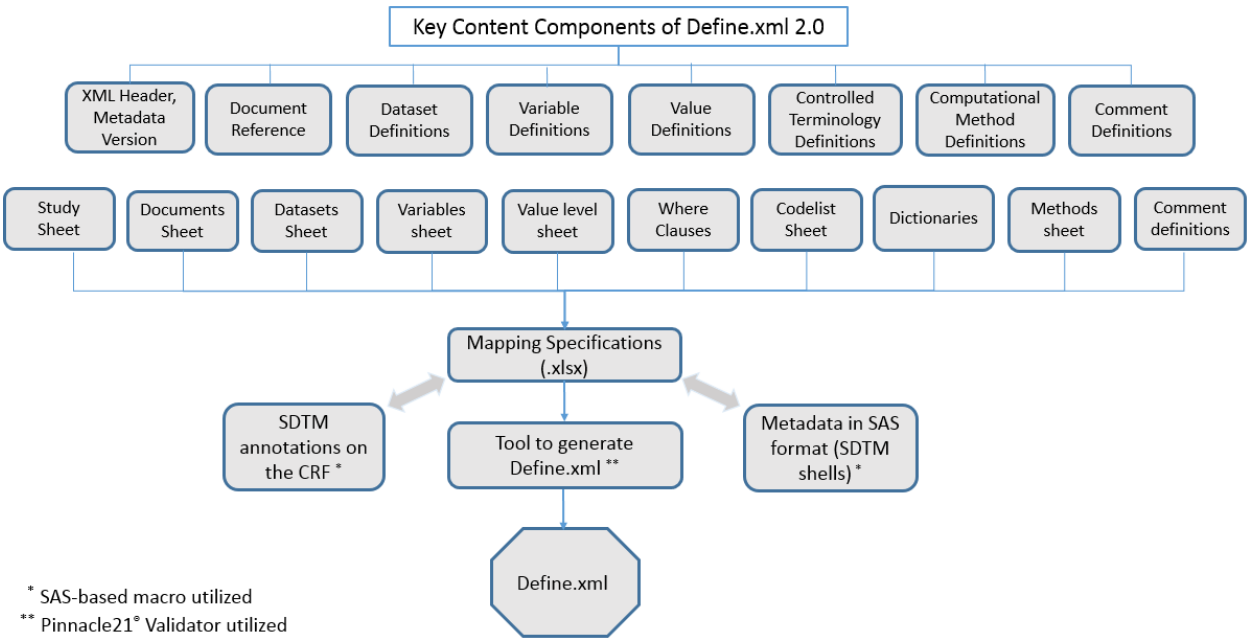

Figure 3. Metadata driven specification components for define.xml generation

SPECIFICATIONS DOCUMENT

All the required metadata specification components are properly arranged into a single file and any additional columns that you may need can be added as and where necessary. The minimum columns that are required to present the metadata information are explained below at the component level.

STUDY SHEET

This sheet provides the information related to study title, protocol number and the version of IG used.

Attribute	Value
StudyName	10001 (PHARMASUG_2016)
StudyDescription	A Phase 3, Double-Blind, Randomized, Vehicle-Controlled, Multicenter, Para
ProtocolName	PHARMASUG_2016_10001

Display 1. Columns from the study sheet

DATASETS SHEET

This sheet provides the dataset metadata and mimics the define.xml datasets sheet. We have added couple of columns for our internal use to provide the programmer information, programming status etc. Additional columns can be added as per our requirements.

Dataset	Description	Class	Structure	Purpose	Key Variables	Source	Programmer	QC Programmer
AE	Adverse Events	EVENTS	One record per adverse e	Tabulation	STUDYID,USUBJID	RAW.AE	psakampally	bbusa
CM	Concomitant Medications	INTERVENTIONS	One record per recorded	Tabulation	STUDYID,USUBJID	RAW.CM	psakampally	bbusa
DA	Drug Accountability	FINDINGS	One record per drug acco	Tabulation	STUDYID,USUBJID	RAW.DA	psakampally	bbusa
DM	Demographics	SPECIAL PURPOSE	One record per subject	Tabulation	STUDYID,USUBJID	RAW.DM RAW	psakampally	bbusa

Display 2. Partial Columns from datasets sheet

VARIABLES (STUDY NOTES) SHEET FOR ALL DOMAINS

This sheet presents the variable metadata information and unlike the general specifications document, all the dataset specifications are placed on the same sheet. There are couple of advantages when the specifications for all the datasets are placed in one single sheet:

Order of Variable	Dataset	Variable	Label	Data Type	Length	Role	Mandatory	Codelist	Origin	Page Number	Study Notes
1	DM	STUDYID	Study Identifier	text	20	Identifier	Yes		Assigned		set to "SW-1001-101-01"
8	DM	RFXSTDTC	Date/Time of First Study Treatment	text	20	Record Qualifier	No		Derived		set to RAW.EX.EXDAT when RAW.EX.VISIT eq "Day 1"
20	DM	RACE	Race	text	40	Record Qualifier	No	RACE	CRF	4	RAW.DM.RACE

Display 3. Sample columns of the variables (specifications) sheet

- Study notes for common variables (USUBJID, XXSEQ…) across domains can be presented in a consistent way by auto-filtering for that variable.
- Attributes for common variables can be maintained same across all datasets easily.
- Process of copying the specs to the define specifications template is easy.
- Determining the order of variable to be presented in the final dataset can be easily adjusted in the template to reflect it in the shell for final dataset.

VALUE LEVEL SHEET

Value level sheet contains the value level metadata information and this sheet resembles to the define specifications sheet. This sheet requires a periodic update while the study is ongoing.

Domain Prefix	Where Variable	Where Comparator	Where Comparator Value	Sourcevariable	Variable Name	Variable Label	Origin	Type	Length
VS	VSTESTCD	EQ	HEIGHT	VSORRES	HEIGHT	Height	CRF	Char	200
VS	VSTESTCD	EQ	WEIGHT	VSORRES	WEIGHT	Weight	CRF	Char	200
VS	VSTESTCD	EQ	BMI	VSORRES	BMI	Body Mass Index	CRF	Char	200
VS	VSTESTCD	EQ	TEMP	VSORRES	TEMP	Temperature	CRF	Char	200

Display 3. Sample columns from value level sheet

WHERECLAUSES

WhereClauses sheet presents the coded and decoded value level metadata information and it resembles the define specifications sheet. This sheet requires a periodic update when the study is ongoing.

ID	Dataset	Variable	Comparator	Value
DISPAMT	DA	DATESTCD	EQ	DISPAMT (Dispensed Amount)
RETAMT	DA	DATESTCD	EQ	RETAMT (Returned Amount)
DCT	FA	FATESTCD	EQ	DCT (Dermatophyte Culture Test)
KOH	FA	FATESTCD	EQ	KOH (KOH Test)

Display 4. Sample columns from WhereClauses sheet

CODELIST SHEET

This sheet and is populated with all possible values for the study. When the study is ongoing the need of creating new formats arises based on the data, in these cases the programmers have to be advised not to write their own formats in program, but instead request the study lead to add that format to the format catalog. This approach minimizes the effort required to update codelist at the end of the study.

ID	Name	NCI Codelist Code	Data Type	Term	NCI Term Code	Decoded Value
RACE	Race	C74457	text	AMERICAN INDIAN OR ALASKA NATIVE	C41259	American Indian or Alaska Native
RACE	Race	C74457	text	ASIAN	C41260	Asian
RACE	Race	C74457	text	BLACK OR AFRICAN AMERICAN	C16352	Black or African American

Display 5. Sample columns of the Codelist sheet

DICTIONARIES SHEET

This sheet presents the dictionaries and their version information that are used in the study.

ID	Name	Data Type	Dictionary	Version
MedDRA	MedDRA Dictionary	text	MedDRA	17.0
WHODrug	WHO Drug Dictionary	text	WHODD	WHODrugB2Enhan

Display 6. Sample columns from the dictionaries sheet

METHODS SHEET

This sheet contains the common derivation/method information that are used across the study. This helps to maintain the same algorithm across the study. Also, it helps in minimizing the errors.

ID	Name	Type	Description
STDY	STUDYDAY	text	STUDYDAY EQ DATEPART (DATEX) - DATEPART(RFSTDTC) +1. IF DATEX LT RFSTDTC then STUDYDAY EQ DATEPART (DATEX) - DATEPART(RFSTDTC).

Display 7. Sample columns from the dictionaries sheet

COMMENTS SHEET

This sheet is used to present the information that should be displayed in the comments column of define.xml. The ID value presented here should match with the comment column of the other sheets described above.

ID	Description	Document	Pages
DM.STUDYID	Value will be "PHARMASUG_2016"		
DM.DOMAIN	Value will be "DM"		
DM.USUBJID	catx('-', STUDYID, SCRNID) Example: PHARMASUG-2016-10001		4

Display 8. Sample columns from the comments sheet

DOCUMENTS SHEET

Documents sheet presents the information of the external documents that are referenced in the Define.

ID	Title	Href
blankcrf	SDTM aCRF	blankcrf.pdf
ReviewersGuide	Reviewers Guide	reviewersguide.pdf
SuppDoc	Supplemental Documentation	supplementaldatadocumentation.pdf

Display 9. Sample columns from the documents sheet

The above worksheets are used to create the final define specifications workbook that can be used to generate the define.xml file using a define generator tool (Pinnacle21 Validator). Any in-house developed tool can also be used, but we have utilized Pinnacle21 validator to generate the define.xml [2].

Once the specifications are finalized, define.xml is generated by passing the specifications through Pinnacle21 validator. Shells for the domains are also generated with the SAS macro and are converted to XPT files. These XPT files are passed through Pinnacle21 for validation along with the define file. Any issues related to the variable attributes, order of variables within domain, missing any FDA expected variables etc. that surface in the define validation report are fixed. CRF annotation can be automated with a SAS macro by using the define file. This approach will significantly reduce the work hours, maintains consistency and decreases the errors.

After fixing the issues the define file and the annotated CRF can be sent to the sponsor/client for review. This approach makes the review process easy when compared to the traditional way of sharing the excel specifications. The advantages of define over excel are:

- All the specifications are in one single XML file which is hyperlinked making the navigation easy from one data point to other with just click of a mouse button.
- Since it is used from the start of the study any issues that are pending can be easily tracked at the end.
- As the CRF is also hyperlinked with the XML file, navigating to the specified place on CRF is made simple.

- Define can be validated for consistency and completeness upfront so that there is no workaround at the end of the study.

CONCLUSION

In this paper, we have demonstrated process to generate define.xml by having the required components of the data definition document built into the dataset mapping specifications. The idea is to generate define.xml upfront and during the CDISC datasets development life cycle. This will benefit the programmers to utilize metadata information stored in the document to validate datasets for compliance against study specific define and at the same time build utility that can automate the generation of SDTM datasets and aCRF. By having this metadata driven approach during the development of the CDISC SDTM and ADaM datasets, the process of generating these datasets will be much more efficient. In addition, as the define.xml is generated upfront, it will be subjected to multiple review cycles by the study team and the client which results in a complete and meaningful define.

REFERENCES

[1] Ron, Fitzmartin and Ginny, Hussong. “Required Electronic Submissions to CDER / CBER.” *fda.gov*. 10-08-2015 Available at http://www.fda.gov/downloads/Drugs/DevelopmentApprovalProcess/SmallBusinessAssistance/UCM467501.pdf

[2] Serigy, Sirichenko; Michael, DiGiantomasso; Travis, Collopy. “Usage of OpenCDISC Community Toolset 2.0 for Clinical Programmers – HT04.” Proceedings of PharmaSUG 2015 Conference

CONTACT INFORMATION

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

Name: Vara Prasad Sakampally
Enterprise: Vita Data Sciences
Address: 281 Winter St. Suite 100
City, State ZIP: Waltham, MA 02451
Work Phone: 270-282-5166
Fax: 781-466-9681
E-mail: psakampally@softworldinc.com
Web: www.softworldinc.com

Name: Bhavin Busa
Enterprise: Vita Data Sciences, Division of Softworld, Inc.
Address: 281 Winter St. Suite 100
City, State ZIP: Waltham, MA 02451
Work Phone: 781-373-8455
Fax: 781-466-9681
E-mail: bbusa@softworldinc.com
Web: www.softworldinc.com