Paper PP15

Automating the blankcrf.pdf and define.xml with your database build

John McDade, PHASTAR, Glasgow, UK

ABSTRACT

At study set-up we receive a raw annotated CRF and database design excel workbook detailing the structure of the raw database with every variable collected on the study, these documents describe the Database Build. The role of these documents are usually confined to aiding the SDTM specification process which then allows us to create the SDTM annotated blankcrf.pdf. What if we could use the database design as the driving force for dynamically creating the define.xml before programming begins?

INTRODUCTION

Database specifications are often an underused asset on a study, these come in excel format and are easy to import to SAS. Although their primary function is to map out the structure of the database in terms of raw datasets and variables, this document can be a key driver in define.xml creation.

The SDTM define.xml is a machine-readable file containing metadata that describes the structure of the SDTM datasets. It contains details of all datasets and variables used, possible codelists and tests used within each domain. The database specification contains much of this detail but at the raw dataset level. This information can be imported and used directly or with an additional mapping step if required.

This paper will explain how the database specifications and raw annotated CRF along with some other key metadata can be used in a process to expedite the creation of SDTM aCRF, mapping specifications and define.xml before SDTM programming. Many companies have moved emphasis to produce define.xml early in the process as there are numerous benefits aside from avoiding critical issues late in the day with a study submission. The process is also dynamic for CDISC controlled terminology.

DATABASE SPECIFICATIONS

Once a database has been built a version of database specifications and a corresponding raw annotated CRF will be released for review. Once reviewed by the necessary teams and agreed this captures all data points required for the study, the database will move into production. If there are subsequent database updates throughout the course of the study, these will be incorporated into new versions of raw CRF and database specifications.

Below is an example DB spec for a raw AE dataset, variables which have the ‘Valid Values’ column populated have associated codelists. The variable AETERM has no values populated as this is a free text field but we can see AEACN has finite values, this is as expected for an Adverse Events page. This data will be imported for all raw variables and used to generate the necessary Value Level Metadata and Codelist information for our define.xml as well as the SDTM primary and supplemental variables.

The database specification below is an example from Medidata Rave®. Although this type of specification is available in excel format for different database vendors, the Rave® raw CRFs can also provide some very useful metadata. We will look at the use of this metadata in the automation of blankcrf section.

Medidata Rave® database specification – raw AE dataset

Variable Name	Variable Label	Type	Length	Collection Length	Format	Valid Values
STUDY	Study Code	C	11	11		
SUBJECT	Enrolment Code	C	8	8		
LINE	Line Number	N	8	3		
AEANY	Any Adverse Events	C	2	2	$NOYES	0=No
			.	.		1=Yes
AENO	AE Number	N	8	3		
AETERM	Adverse Event	C	200	200		
AESTDAT	AE Start Date	C	10	10		
AEENDAT	AE Stop Date	C	10	10		
AESER	**Serious AE**	C	2	2	$NOYES	0=No
			.	.		1=Yes
AEACN	Action Taken, Investigational Product	C	2	2	$AEACN	0=Dose Not Changed
			.	.		1=Dose Increased
			.	.		2=Dose Reduced
			.	.		3=Drug Interrupted
			.	.		4=Drug Permanently Discontinued
			.	.		98=Not Applicable
AEREL	Reasonable Possibility AE Caused by IP	C	2	2	$NOYES	0=No
			.	.		1=Yes

CODELISTS AND VALUE LEVEL METADATA

The challenge in creating high quality SDTM specifications and define.xml is not in which variables may or may not be present – many companies automate generic specifications to this point which still require much manual intervention. The challenge is automating study specific codelists and value level metadata which is the most labour intensive part of define.xml creation and most poorly executed [1] [2].

Some common problems with codelists include:

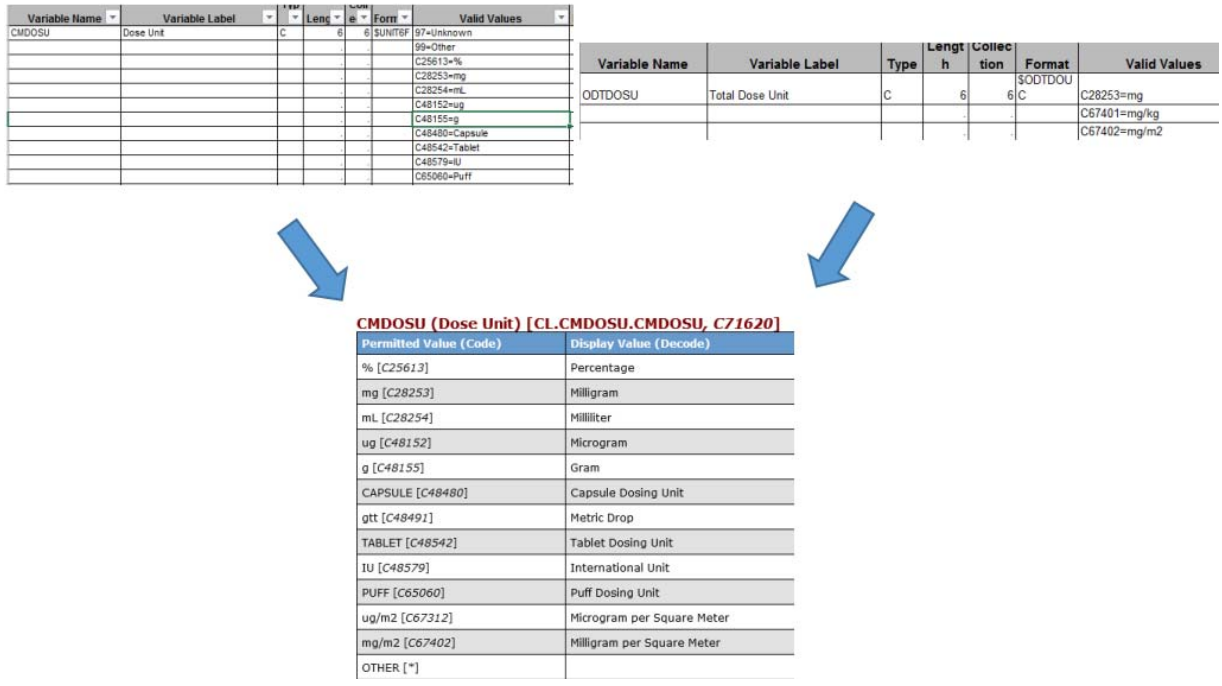
- Codelist completely missing when relevant
- Shared codelists across domains when should be specific to domain e.g. for xxORRESU
- Dumping of entire NCI codelist into define.xml

Value level metadata can also be problematic, from the CDISC Define-XML Specification Version 2.0 [3]. "**It is left to the discretion of the creator when it is useful to provide Value Level Metadata and when it is not. The creator should speak to the consumer if they would like further guidance**." While such a statement leaves flexibility for the creator and consumer, it can lead to inconsistencies in approach by different study teams and also cause a headache when trying to automate such metadata.

CODELIST EXAMPLE

The variables CMDOSU and ODTDOSU below are from distinct raw datasets but both feed into the SDTM variable CMDOSU. The Valid Values column are the possible values when data is entered in this field. We can easily extract this information and combine to build our CMDOSU codelist. The same process will be followed for all variables with codelists.

Note that the NCI codes are included with this database specification, this is not always the case. As we have the terms used and know which parent codelist the terms must sit under, we can merge to a specific version of the NCI controlled terminology to obtain the relevant NCI codes. In practice this is supplied as a macro parameter.



Variable Name	Variable Label	Typ	Leng	e	Form	Valid Values
CMDOSU	Dose Unit	C	6	6	$UNITSF	97=Unknown
						99=Other
						C25613=%
						C28253=mg
						C28254=mL
						C48152=ug
						C48155=g
						C48480=Capsule
						C48542=Tablet
						C48579=IU
						C65060=Puff

Variable Name	Variable Label	Type	Length	Collection	Format	Valid Values
ODTDOSU	Total Dose Unit	C	6	6	$ODTDOU C	C28253=mg
						C67401=mg/kg
						C67402=mg/m2

CMDOSU (Dose Unit) [CL.CMDOSU.CMDOSU, *C71620*]

Permitted Value (Code)	Display Value (Decode)
% [*C25613*]	Percentage
mg [*C28253*]	Milligram
mL [*C28254*]	Milliliter
ug [*C48152*]	Microgram
g [*C48155*]	Gram
CAPSULE [*C48480*]	Capsule Dosing Unit
gtt [*C48491*]	Metric Drop
TABLET [*C48542*]	Tablet Dosing Unit
IU [*C48579*]	International Unit
PUFF [*C65060*]	Puff Dosing Unit
ug/m2 [*C67312*]	Microgram per Square Meter
mg/m2 [*C67402*]	Milligram per Square Meter
OTHER [*]	

VALUE LEVEL EXAMPLE

Although mentioned previously that the extent of value level metadata provided can be at the discretion of the creator, there are some minimum requirements expected and these can be used in the basis of automation. For findings domains, the value of xxORRES should be given in terms of xxTESTCD where different metadata attributes are expected per test code value. The other minimum requirement is the SUPP variables described in terms of QNAM value.

Taking the Vital Signs domain as an example, below is a list of possible tests in the DB spec. These can be extracted and will be used for a dual purpose, this will generate value level information and also the VSTEST and VSTESTCD codelists.

Variable Name	Variable Label	Typ	Length	Collection Length	Forma	Valid Values
VSTEST	Test	C	7	7	$VSTEST	C25206=Temperature
						C25208=Weight
						C25298=Systolic Blood Pressure
						C25299=Diastolic Blood Pressure
						C49676=Pulse Rate
						C49678=Respiratory Rate

Value Level Metadata - VS [VSORRES]

Variable	Where	Type	Length / Display Format	Controlled Terms or Format	Origin	Derivation/Comment
VSORRES	VSTESTCD EQ TEMP (Temperature)	text	200		CRF Pages 28 31 35 43	
VSORRES	VSTESTCD EQ WEIGHT (Weight)	text	200		CRF Pages 28 31 35 43	
VSORRES	VSTESTCD EQ SYSBP (Systolic Blood Pressure)	text	200		CRF Pages 28 31 35 43	
VSORRES	VSTESTCD EQ DIABP (Diastolic Blood Pressure)	text	200		CRF Pages 28 31 35 43	
VSORRES	VSTESTCD EQ HEIGHT (Height)	text	200		CRF Pages 28 31 35 43	
VSORRES	VSTESTCD EQ PULSE (Pulse Rate)	text	200		CRF Pages 28 31 35 43	
VSORRES	VSTESTCD EQ RESP (Respiratory Rate)	text	200		CRF Pages 28 31 35 43	

VSTEST and VSTESTCD codelists from VS test information:

Variable Name	Variable Label	Typ	Length	Collection Length	Form	Valid Values
VSTEST	Test	C	7	7	$VSTEST	C25206=Temperature
						C25208=Weight
						C25298=Systolic Blood Pressure
						C25299=Diastolic Blood Pressure
						C49676=Pulse Rate
						C49678=Respiratory Rate

Vital Signs Test Name [CL.VSTEST, *C67153*]

Permitted Value (Code)
Temperature [*C25206*]
Weight [*C25208*]
Systolic Blood Pressure [*C25298*]
Diastolic Blood Pressure [*C25299*]
Height [*C25347*]
Pulse Rate [*C49676*]
Respiratory Rate [*C49678*]

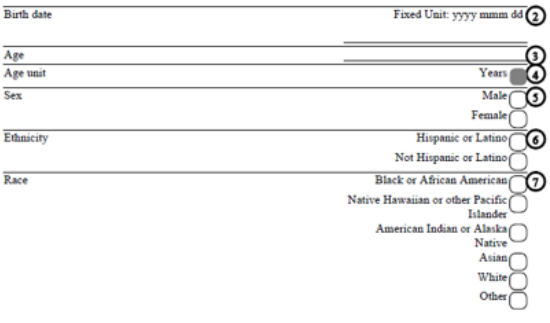
Vital Signs Test Code [CL.VSTESTCD, *C66741*]

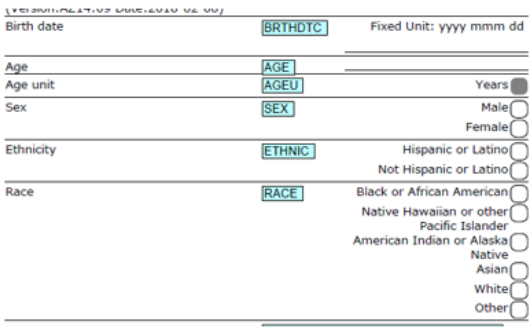
Permitted Value (Code)	Display Value (Decode)
TEMP [*C25206*]	Temperature
WEIGHT [*C25208*]	Weight
SYSBP [*C25298*]	Systolic Blood Pressure
DIABP [*C25299*]	Diastolic Blood Pressure
HEIGHT [*C25347*]	Height
PULSE [*C49676*]	Pulse Rate
RESP [*C49678*]	Respiratory Rate

BLANKCRF AUTOMATION

Automation of the blankcrf is a very useful tool as such a manual process can be extremely tedious and not a highly desirable task for many programmers, especially on longer CRFs. Creating the aCRF programmatically also allows dynamic changes of colours, fonts, positioning if required.

The CRF on the left below has hyperlinks from which we can extract their y-coordinate on the PDF page. Combining this with an allocated x-coordinate, page numbers and the SDTM annotation can build an FDF file which is simply imported on the blank CRF.





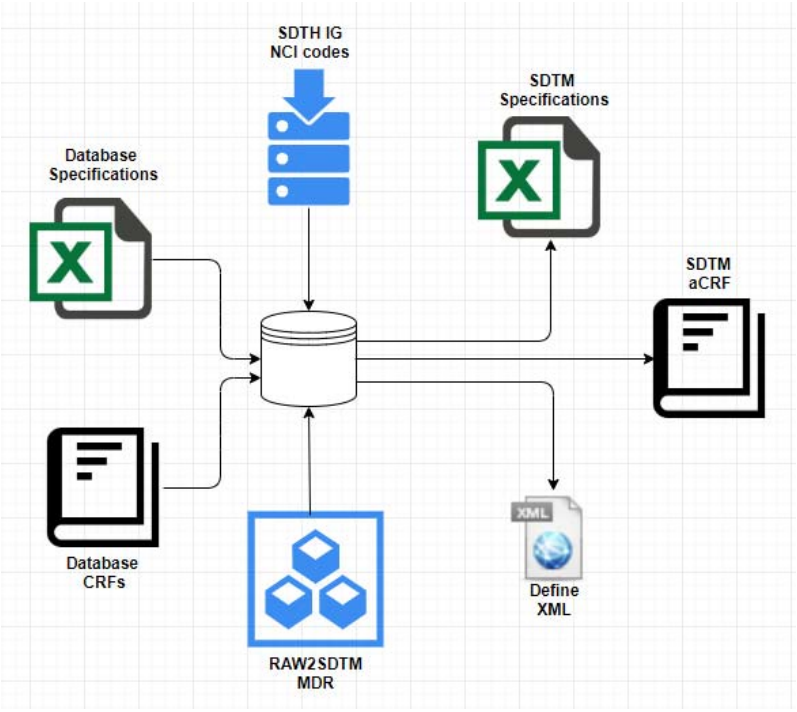
One of the key benefits of this process is that the page numbering is not just used in building the FDF file for the aCRF, this is also used in the CRF page origins for the define.xml as below.

BRTHDTC	Date/Time of Birth		datetime		ISO8601	CRF Page 10
AGE	Age		integer	8		CRF Page 10
AGEU	Age Units		text	10	["YEARS" = "Year"] <Age Unit>	CRF Page 10
SEX	Sex		text	1	["F" = "Female", "M" = "Male"] <Sex>	CRF Page 10
RACE	Race		text	60	Race	CRF Page 10
ETHNIC	Ethnicity		text	60	["HISPANIC OR LATINO" = "Hispanic or Latino", "NOT HISPANIC OR LATINO" = "Not Hispanic or Latino"] <Ethnic Group>	CRF Page 10

SUMMARY

An overview of the full process is given below. The database specifications and database CRFs are the study specific components which will change. The CDISC metadata repository (MDR) contains both SDTM IG information and historical Controlled Terminology versions. The RAW2SDTM MDR is simply a dataset which links the raw variables and the SDTM counterpart. Many clients can provide this from their internal mapping standards which ensures consistency of mapping across studies and vendors.

It is important to note that the define.xml in this process is generated prior to SDTM programming and a set of SDTM specifications can be created at the same time. Default attributes can be given to variables at this point and retrospectively updated based on the actual data on study completion as it serves no purpose to update length fields while data collection is ongoing.



CONCLUSION

In conclusion, the paper demonstrates the use of database specifications and raw annotated CRF in expediting key components of the CRT package. The process is dynamic to a nominated controlled terminology and is of particular benefit when pooling studies. With the tedious but essential tasks of annotating the CRF and building codelists taken care of automatically, programmers can concentrate on the actual study programming.

REFERENCES

1. David Roulstone, Pinnacle 21. (2018). *Do's and Don'ts of Define.xml*. [online] Available at: https://www.lexjansen.com/phuse/2018/sa/SA04.pdf [Accessed 21 Sep. 2019].
2. Pinnacle 21. (2019). *Do's and Don'ts of Define.xml Webinar Q&A*. [online] Available at: https://www.pinnacle21.com/blog/dos-donts-definexml-webinar-qa [Accessed 21 Sep. 2019].
3. CDISC Define-XML Specification Version 2.0 (2013) Available at: https://www.cdisc.org/

CONTACT INFORMATION

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

John McDade
PHASTAR
Tara House, 46 Bath Street,
Glasgow G2 1HG
Email: john.mcdade@phastar.com
Web: https://phastar.com/