Define.xml validation – SAS base solutions

Sam Tomioka, Sunovion Pharmaceuticals Inc, Fort Lee, USA
Steven Huang, Sunovion Pharmaceuticals Inc, Fort Lee, USA

ABSTRACT

Due to lack of commercially available tools for the validation of the contents of the *define.xml*, and the increasing submission activities, we at Sunovion had developed our in-house programs to address the common issues/mistakes of the define files. This article will illustrate the systematic approach of using SAS programs to validate the contents of the define.xml. We will demonstrate how the source data are collected; explain the process of validation including a graphical overview and most common findings from the validation process.

INTRODUCTION

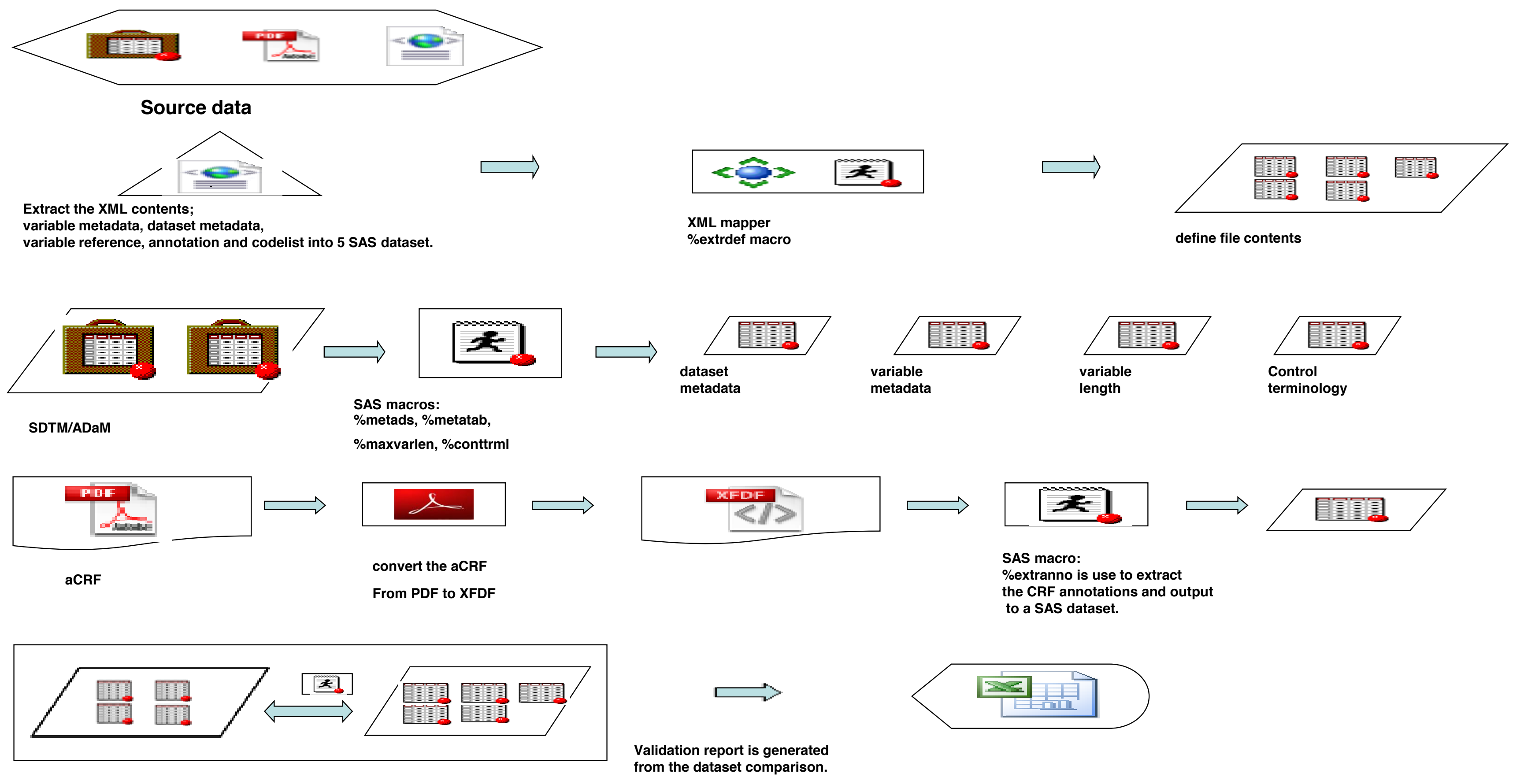
Define.xml is a data definition file, formally known as CRT DD, it is use to facilitate the review of the study data submitted to the regulatory agency. A well prepare standardized metadata can minimized the time require for the reviewer to be familiarize with the data, which can speed up the overall review process.

In December 2011, the FDA published "CDER Common Data Standards Issues Document" to address the commonly observed issues in the standardized data submission. The document stated that "sponsors should make certain that every data variable's codelist, origin, and derivation is clearly and easily accessible from the define file". Furthermore, the FDA listed "Define Doesn't Validate" as one of the common errors that they have observed.

There are some validation tool which checks for conformance to the XML schema and ODM specifications that addresses the issue of "Define Doesn't Validate", however these tools does not validate the contents of the define file. SAS Clinical Standard Toolkit (Ver.1.4) and OpenCDISC were available to us, however SAS/CST only validates the structure and syntax of the define file against the XML schema (define1-0-0.xsd; CRT-DDS standard), and OpenCDISC uses default CDISC Define.xml 1.0 Validation Rules to checks for ODM extensions and the consistency of data types defined for variables, values, and codelist name.

To ensure accuracy, consistency, and completeness of the define.xml, we have developed our in-house tool to supplement the checks such as dataset level metadata, variable level metadata (including the CRF pages), value level metadata, and the controlled terminology that are not covered under SAS/CST and OpenCDISC.

OVERVIEW:



COMPARSION OUTPUT:

These columns come from define.xml i.e. output from %extrdef

These columns come from annotated CRF i.e. output from %extranno

Define OID	domain	var	pa	anno	listed
	AE	AEENRF	104	AE.AEENRF	aCRF
CF.CFDTC	CF	CFDTC	3		Define
CF.CFTEST	CF	CFTEST	30		Define
	CM	CMENRF	100	CM.CMENRF	aCRF
DA.DADTC	DA	DADTC	3		Define
	DA	DAORRES	98	DA.DAORRES	aCRF
DA.EPOCH	DA	EPOCH	97		Define
	DA	VISIT	97	DA.VISIT	aCRF
DM.DMDTC	DM	DMDTC	3		Define
	EG	EGMETHOD	35	EG.EGMETHOD	aCRF
	EX	EXLOT	97	EX.EXLOT	aCRF
	HO	HOCAT	103	HO.HOCAT = 'PSYCHIATRIC HOSPITALIZATION'	aCRF
	HO	HOTERM	103	HO.HOTERM='HOSPITALIZATION'	aCRF
IE.IEDTC	IE	IEDTC	3		Define

AEENRF is annotated on page 104 of blankcrf.pdf, however, the define OID is missing, therefore the Origin of variable in define.xml does not come from page 104.

define.xml defines the variable CFDTC is annotated on page 3 of blankcrf.pdf, however, page 3 of the blankcrf.pdf is missing the annotation.

Finding from the SDTM dataset:

Domain Name	Variable Name	Error Type	Value in Define	Expected Value	Comment
LB	LBTEST	Control Terminology	2	LDL	Control Terminology in Define.xml are not consistant.
AE	AERELN	Control Terminology	Not Listed		Control Terminology Missing in Define.xml
DS	DSDECOD	Control Terminology	Adverse Events	Not Listed	Control Terminology Missing in dataset
QS	QSSCAT	Value Level Metadata	Not Listed		This should be added to define.xml
QS	QSSCAT	Value Level Metadata	Not Specified	Not Listed	This is added to define.xml but not present in datasets
QS	QSTESTCD	Value Level Metadata	text	float	Datatype Specified in define.xml and datasets are not consistant.
LB	LBTESTCD	Value Level Metadata	text	float	Datatype Specified in define.xml and datasets are not consistant
VS	VSTESTCD	Value Level Metadata	text	float	Datatype Specified in define.xml and datasets are not consistant
LB	LBTESTCD	Value Level Metadata	Hemoglobin	HEMOGLOBIN	Value Label Specified in define.xml and datasets are not consistant

Findings from the analysis datasets:

Domain Name	Variable Name	Error Type	Value in Define	Expected Value	Comment
ISSAIMS	QSTEST	Value Level Metadata	Text	float	Data type Specified in define.xml and datasets are not consistent
ISSAIMS	QSTEST	Value Level Metadata	Current Problems with Teeth,Dentures	Current Problems With Teeth, Dentures	Value Label Specified in define.xml and datasets are not consistent
ISEPANS	QSTEST	Value Level Metadata	Mannerisms and Posturing	Mannerisms And Posturing	Value Label Specified in define.xml and datasets are not consistent
ISSBT	LBTESTCD	Value Level Metadata	C-telopeptide	SERUM C-TELOPEPTIDE COLLAGEN TYPE 1	Value Label Specified in define.xml and datasets are not consistent
ISSVS	VSPOS	Value Level Metadata	Missing Value	Not Listed	This is added to define.xml but not present in datasets
ISSCHEM	ABLFL	Control Terminology	1	Not Applicable	Control Terminology is duplicated in Define.xml
ISSCHEM	LBTESTCD	Value Level Metadata	ASPARTATE AMINOTRANSFERASE	AST	Value Label Specified in define.xml and datasets are not consistent
ISSCHEM	COUNTRY	Control Terminology	ARG	Not Applicable	Control Terminology is duplicated in Define.xml
ISSCHEM	LBTESTN	Control Terminology	Not Listed	19	Control Terminology missing in define.xml
ISSAE	AERELN	Control Terminology	Not Listed		Control Terminology Missing in Define.xml

SOURCE CONTENT EXTRACTION

The contents within the define file consists of SDTM annotated CRF, specification documents, and the SAS transport files. We will demonstrate the approach that we took to compile the source contents of the define.xml and provide a detail process on how the contents are extracted.

DEFINE.XML:

A XML map file is created to provide instruction to the SAS libname engine on how to extract metadata within the define.xml into the SAS datasets.
Below is an example of the XML map file:

```
<TABLE name="metadata">
<TABLE-DESCRIPTION>Metadata of variables</TABLE-DESCRIPTION>
<TABLE-PATH syntax="XPath">/ODM/Study/MetaDataVersion/ItemDef</TABLE-PATH>

<COLUMN name="def_oid">
<PATH syntax="XPath">/ODM/Study/MetaDataVersion/ItemDef/@OID</PATH>
<DESCRIPTION>Define OID</DESCRIPTION>
<TYPE>character</TYPE>
<DATATYPE>string</DATATYPE>
<LENGTH>1000</LENGTH>
</COLUMN>
```

Define the name of the resultant dataset.

Define the path within the define.xml for the collections of records with a defined set of columns for SAS dataset.

Define variable attributes such as: variable name, description, data type, and length.

SDTM ANNOTATED CRF:

Using Adobe Acrobat 9 Pro version, annotations and form data within a blankcrf.pdf can be extracted as **X**ML **F**orms **D**ata **F**ormat. With the SAS XML engine, this XFDF can be converted to a SAS dataset.
Below is an example of the XFDF file:

```
<annots>
<freetext width="1.5" color="#FFFFFF" creationdate="D:00000000000000Z" flags="print"
  date="D:20101206102547-05'00'" name="e2085b08-e2f8-4c01-a2b7-f86230ec3743" page="2"
  justification="centered" rect="164.093994,565.987976,223.391006,580.072998" title="T">

<contents-richtext>
  <body xmlns="http://www.w3.org/1999/xhtml" xmlns:xfa=http://www.xfa.org/schema/xfa-data/1.0/
    xfa:APIVersion="Acrobat:9.2.0" xfa:spec="2.0.2" style="font-size:8.0pt;text-
    align:left;color:#0000FF;font-weight:normal;font-style:normal;font-family:Arial;font-stretch:normal">
       <p dir="ltr">SV.SVSTDTC</p>
   </body>
 </contents-richtext>

 <defaultappearance>0 0 1 rg /ArialMT 8 Tf</defaultappearance>
 <defaultstyle>font: Arial 8.0pt; text-align:left; color:#0000FF </defaultstyle>
</freetext>
```

Form page number; starts from zero, generated by Acrobat.

The text of the annotation are contain within the <contenst-richtext>.

Actual text.

The style of the annotation text can be define using <span> <span> within the "p" tag.

The **'annotation'** dataset is created with the **%extranno** macro and the above XML map file.
It consist of three variables: **'annotation1'(<p> tag)**, **'annotation2'(<span> tag)**, and **'page'**.
Additional post process require the concatenation of annotation1 and annotation2, and the renumbering of the first page from page 0 to page 1. The domain and variable name are derived from the annotations which will be the key variables for the comparison process.

SAS TRANSPORT FILES:

The extraction of the metadata from SAS transport files are done as follow:
1).**%xpt2sas** macro is to convert SAS transport files into SAS datasets, then the DICTIONARY SAS data views were used to extract the dataset metadata and variable metadata.
2).**%metads** macro is used to create DOMAIN dataset which contains dataset metadata and **%metatab** macro is used to create COLUMN dataset which consists of variable metadata.
3).**%getvlm** macro is used to generate value level metadata for –TESTCD variables and QNAM variables
4).**%gtvlmtyp** macro is used to identify the data type between parent variable level and value levels.
5).**%maxvarlen** macro is used to derive the length of variables as well as the length at value levels.
6). **%conttrm1** macro is used to generate a list of the control terminology from the SAS dataset then it will compare the list with our standard metadata specifications which defines the controlled terminology that the variable uses.

VALIDATION AND VERIFICATION PROCESS:

Our content validation checks can be categorized as follows:

1) **Dataset Metadata:** Standard dataset metadata from sponsor's defined standard (DSP-SDTM) was compared against the **'table_meta'** dataset from **%extrdef** macro for the following items:
   a) Domain Purpose (standard vs define.xml).
   b) Domain Structure (standard vs define.xml).
   c) Reference Data (standard vs define.xml).
   d) Dataset Descriptions (standard vs define.xml).
   e) Repeating Data (standard vs define.xml).
   f) Class Data (standard vs define.xml).

2) **Variable Metadata:** Datasets '**metadata**', '**var_ref**', and '**annot**' from **%extrdef** macro were compared against the outputs from **%metatab** and **%extranno** macros for the following items:
   a) Data Type (define.xml vs submission datasets).
   b) Variable Label (define.xml vs submission datasets).
   c) Consistency of variables (define.xml vs submission datasets) - Identifies variables missing in define.xml or XPT
   d) Variable Length (define.xml vs submission datasets) - Checks for maximum length defined in Define.xml with actual maximum length in XPT.
   e) Variable Order (define.xml vs submission datasets).
   f) Origin (define.xml vs annotated CRF).

3) **Controlled Terminology:** output from **%conttrm1** macro was compared against '**codelist**' dataset from **%extrdef** macro for the following items:
   a) Checks for duplication (define.xml).
   b) Consistency checks (define.xml vs submission datasets).
   c) Missing controlled terminologies in define.xml or datasets.

4) **Value Level Metadata:** output from **%conttrm1** was compared against **'codelist'** dataset from **%getvlm** for the following items:
   a) Check for incorrect value level metadata (submission dataset vs define.xml).
   b) Check for consistency (submission dataset vs define.xml).
   c) Missing Value Level Metadata in define.xml.
   d) Identify Value Level data that does not exist in the submission dataset.
   e) Data Type (submission dataset vs define.xml).
   f) Variable Label (define.xml vs submission datasets).

CONCLUSION

It is important to ensure the accuracy, consistency, and completeness of the define.xml prior to the submission. However it is a challenging process to check the define.xml against the source metadata or submission datasets with existing tools. With the use of our validation tool most of the tedious and time consuming checks such as the dataset level metadata, variable level metadata, value level metadata, and the controlled terminology in the define.xml against the actual submission datasets can now be accomplished quickly and precisely. Programmers with solid SAS and submission knowledge can easily understand and utilize the tool. Furthermore our simplified output format can clearly point out the discrepancy which enable the issues to be easily identified.

REFERENCES

CDER Common Data Standards Issues Document Version 1.1/December 2011
http://www.fda.gov/downloads/Drugs/DevelopmentApprovalProcess/FormsSubmissionRequirements/ElectronicSubmissions/UCM254113.pdf
Case Report Tabulation Data Definition Specification Final Version 1.0
http://www.cdisc.org/stuff/contentmgr/files/0/464923b10ea16b477151fcaa9f465166/misc/crt_ddspecification1_0_0.pdf
XML Schema Validation for Define.xml White Paper
http://www.cdisc.org/stuff/contentmgr/files/0/464923b10ea16b477151fcaa9f465166/misc/definereport_v1_0.pdf
SAS Clinical Standards Toolkit 1.4: User's Guide
http://support.sas.com/documentation/cdl/en/clinstdtktug/64439/PDF/default/clinstdtktug.pdf
SAS(R) 9.2 XML LIBNAME Engine: User's Guide, Second Edition
http://support.sas.com/documentation/cdl/en/engxml/62845/PDF/default/engxml.pdf
XML Forms Data Format Specification, version 3 August 2009 http://partners.adobe.com/public/developer/en/xml/XFDF_Spec_3.0.pdf

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:
Email: sam.tomioka@sunovion.com or steven.huang@sunovion.com

SUNOVION