PharmaSUG China 2016 – Paper 77

Define.xml Content Validation – CRF Page Check

Xianhua (Allen) Zeng, PAREXEL International, Shanghai, China
Shenglin Zhang, PAREXEL International, Shanghai, China

ABSTRACT

The FDA 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". There are some validation tools that can be used to validate the define.xml, such as SAS Clinical Standard Toolkit and Pinnacle 21 Community. However, these tools do not validate the contents of the define file, it is important to supplement it with other validation checks to ensure the accuracy of your define.xml. To do these additional validations manually can be cumbersome and time consuming. This paper presents a SAS macro ChkDefCrfPage, which extracts define.xml contents, annotations and page information from blankcrf.pdf into SAS datasets using XMLMap and then does the CRF page number consistency check between define.xml and annotated CRF.

INTRODUCTION

The purpose of this paper is to introduce a SAS based approach for checking CRF page number inconsistencies between the define.xml and annotated CRF. The following figure gives a high level overview of the basic concept of this paper.

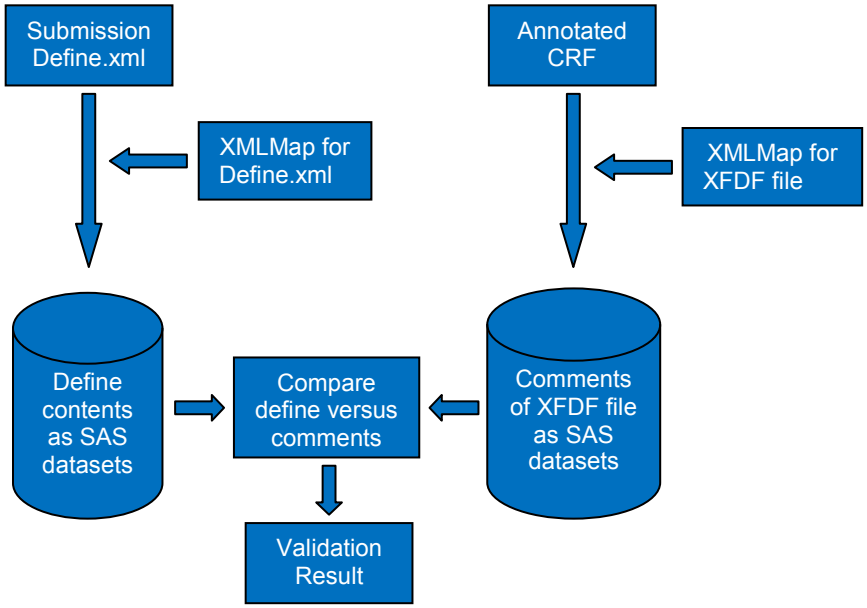

Figure 1. Overview of Basic Paper Concept

The XMLMap extracts metadata from define.xml and annotations from the XML Forms Data Format file exported from blankcrf.pdf. The paper briefly explains how this transformation can be done. The XMLMap can be created using SAS XML mapper. The technique of creating that map is beyond the scope of this paper. This technique is discussed in greater details in Wendi L. Wright's paper at NESUG 2010. The content of XFDF file can be extracted into SAS datasets using the same methodology to extract define.xml. Note that the code for this paper was written in UNIX operating environment. Therefore, developers would need to make some adjustments to the code based on their operating environment.

MACRO PARAMETERS

We call this macro as %ChkDefCrfPage and it requires 3 input parameters as listed in Table 1 below:

Parameter	Description
DEFCRFPATH	Path where define.xml to be validated and annotated CRF are located

XMLMAPPATH	Path where XMLMap is located
OUTPATH	Path where check result is located

Table 1. ChkDefCrfPage Parameters

DEFINE.XML CONTENT EXTRACTION

This extraction is possible using XMLMap and XMLV2 engine of SAS. The first step here is to create XMLMap for define.xml using SAS XML Mapper. The XMLMap tells the SAS XML engine how to map the content in the hierarchical define.xml file to rows and columns in the rectangular SAS tables. Once we have XMLMap, the following code can transform contents of define.xml into SAS datasets.

```
filename define "&defcrfpath/define.xml";
filename defmap "&xmlmappath/define2sas.map";
libname  define xmlv2 xmlmap=defmap automap=replace access=readonly;

proc copy in=define out=work;
run;
```

The datasets that may be created when you use your version of XMLMap, may have a different structure and/or names. The key is to understand the structure and primary key of datasets created using XMLMap. Once you understand it, then you can use your own programming style to create datasets that you can work with.

ANNOTATED CRF COMMENTS EXTRACTION

Annotations and form data within a SDTM annotated CRF (i.e. blankcrf.pdf) was extracted as XML Forms Data Format (XFDF) using Adobe Acrobat. The steps to export the annotation data vary with different Acrobat version. The Acrobat version 7.0 has a very simple process.

- Select “Comments → Export Comments → To File” from the main menu.
- A new box with the title “Export Comments” appears. Save this resulting PDF as XFDF file at your location. Select save as type to be “Acrobat XFDF Files (*.xfdf)”.

The content of XFDF file can be extracted into SAS datasets using the same methodology to extract define.xml.

```
filename blankcrf "&defcrfpath/blankcrf.xfdf";
filename acrfmap "&xmlmappath/acrf2sas.map";
libname  blankcrf xmlv2 xmlmap=acrfmap automap=replace access=readonly;

proc copy in=blankcrf out=work;
run;
```

PARSING IMPORTED COMMENTS WITH PERL REGULAR EXPRESSION

SAS programmers who have ever tried to parse dynamic data to match patterns using traditional functions like INDEX, SCAN and SUBSTR know that creating these statements can become confusing and laborious even when trying to match patterns with low relative complexity. While complex Perl regular expression can also be confusing even for those well experienced with them, the flexibility and power they provide easily make up for the effort required to implement them.

In this section, the ultimate goal is to parse the comments to identify the variable. The code used to do this task is shown below.

```
/*Create variables list*/
proc sql noprint;
    select distinct VARNAME into :varlist separated by "|"
        from define
        ;
quit;

/*Extract variable name from &varlist*/
data comments03;
    set comments02;
    retain REX;
    if _N_=1 then REX=prxparse("s/.*?(\b(?:&varlist)\b)?/\1 /");
    COMMENTS=cats(compbl(prxchange(REX, -1, cats(COMMENTS))));
```

```
    if not missing(COMMENTS);
run;
```

Here's a brief explanation of the regular expression used in the example above. The "." matches any single character except newline. The "*?" is lazy repetition factor, matches 0 or more occurrences of the preceding character as few times as possible. The first "(" and ")" characters matches a pattern and creates a capture buffer for the match. The last "?" is greedy repetition factor, matches the first capturing group zero or one time as many times as possible. The "\b" matches a word boundary. Since we want to mention "\b" only once, so the second "(" and ")" characters are needed. The "(?:…)" means non-capturing group, the "?:" is not necessary in this example. Since there is no memory required for the second catch (?:), it may work faster. The "\1" matches capture buffer 1.

MACRO SOURCE CODE

```
%macro ChkDefCrfPage(defcrfpath =
                   , xmlmappath =
                   , outpath    =
                     );

/*Prevent multi-threaded sorting and message to the SAS log about the maximum length
for strings in quotation marks*/
options NOTHREADS NOQUOTELENMAX;

/*Read define.xml into SAS dataset*/
filename define "&defcrfpath/define.xml";
filename defmap "&xmlmappath/def2sas.map";
libname  define xmlv2 xmlmap=defmap automap=replace access=readonly;

proc copy in=define out=work;
run;

/*Create variable metadata*/
proc sql;
    create table vardef as
        select ITEMOID, ORIGIN
        from define.itemref1 a
        left join
        define.itemdef b
        on a.ITEMOID=b.OID
        ;
quit;

/*Derive the CRF page*/
data define;
    set vardef;
    length DOMAIN $2 VARNAME $8 PAGE_DEF $500;
    if not prxmatch("/^SUPP/", cats(ITEMOID)) and prxmatch("/CRF/o", ORIGIN);
    DOMAIN=scan(ITEMOID, 1, ".");
    VARNAME=scan(ITEMOID, 2, ".");
    PAGE_DEF=cats(prxchange("s/[a-z]+,*|,\s*[a-z]+//io", -1, cats(ORIGIN)));
    PAGE_DEF=cats(prxchange("s/,\s*/, /io", -1, cats(PAGE_DEF)));
    keep DOMAIN VARNAME PAGE_DEF;
    proc sort nodupkey;
    by DOMAIN VARNAME PAGE_DEF;
run;

/*Read comments of blankcrf.pdf into SAS dataset*/
filename blankcrf "&defcrfpath/blankcrf.xfdf";
filename acrfmap "&xmlmappath/acrf2sas.map";
libname  blankcrf xmlv2 xmlmap=acrfmap automap=replace access=readonly;

proc copy in=blankcrf out=work;
run;
```

```
/*Combine the annotations and delete unwanted record*/
data comments01;
    merge p span;
    by P_ORDINAL;
    length COMMENTS COMMENTSL $32767;
    COMMENTS=catx(" ", P, SPAN);
    COMMENTS=compress(COMMENTS, , "kw");
    COMMENTSL=lag(COMMENTS);
    if countc(COMMENTS, "=")=1 then COMMENTS=scan(COMMENTS, 1, "=");
    if prxmatch("/^(note|note:|example|example:|:|\d+\.)$/io", cats(COMMENTSL)) then
COMMENTS="Note: "||cats(COMMENTS);
    if not prxmatch("/^(note|example)/io", cats(COMMENTS));
    FREETEXT_ORDINAL=BODY_ORDINAL;
run;

/*Derive CRF page*/
data comments02;
    merge comments01 freetext;
    by FREETEXT_ORDINAL;
    PAGE=PAGE+1;
    keep COMMENTS PAGE;
run;

/*Create variables list*/
proc sql noprint;
    select distinct VARNAME into :varlist separated by "|"
        from define
        ;
quit;

/* Extract variable name from &varlist*/
data comments03;
    set comments02;
    retain REX;
    if _N_=1 then REX=prxparse("s/.*?(\b(?:&varlist)\b)?/\1 /");
    COMMENTS=cats(compbl(prxchange(REX, -1, cats(COMMENTS))));
    if not missing(COMMENTS);
run;

/*Split COMMENTS having multiple delimiters " "*/
data comments03;
    set comments03;
    length VARNAME $8;
    I=1;
    if prxmatch("/\s/", cats(COMMENTS)) then do until(scan(COMMENTS, I, " ")="");
        VARNAME=cats(scan(COMMENTS, I, " "));
        output;
        I+1;
    end;
    else do;
        VARNAME=COMMENTS;
        output;
    end;
run;

/*Derive DOMAIN*/
data comments03;
    length DOMAIN $2 VARNAME $8;
    set comments03;
    if prxmatch("/SUBJID|DTHDTC|BRTHDTC|AGE|AGEU|SEX|RACE|ETHNIC/o", VARNAME) then
DOMAIN="DM";
    else DOMAIN=substr(VARNAME, 1, 2);
    keep DOMAIN VARNAME PAGE;
```

```
    proc sort nodupkey;
    by DOMAIN VARNAME PAGE;
run;

/*Combine the crf page*/
data acrf;
    set comments03;
    by DOMAIN VARNAME;
    length PAGE_CRF $500;
    retain PAGE_CRF ;
    if first.VARNAME then PAGE_CRF=cats(PAGE);
    else PAGE_CRF=catx(", ", PAGE_CRF, cats(PAGE));
    if last.VARNAME;
    drop PAGE;
run;

/*Check*/
data ChkDefCrfPage;
    merge define acrf;
    by DOMAIN VARNAME;
    if PAGE_DEF^=PAGE_CRF;
    label DOMAIN   = "Domain Abbreviation"
          VARNAME  = "SDTM Variable"
          PAGE_DEF = "CRF Page in Define.xml"
          PAGE_CRF = "CRF Page in Annotated CRF"
          ;
run;

/*Produce validation report*/
ods path work(update) sashelp.tmplmst(read);

ods listing close;
ods tagsets.excelxp file="&outpath/ChkDefCrfPage_%sysfunc(date(),yymmddn8.).xls"
                    options(embedded_titles       = "yes"
                            autofilter            = "all"
                            orientation           = "landscape"
                            sheet_name            = "ChkDefCrfPage"
                            autofit_height        = "yes"
                            print_footer          =  " "
                            row_repeat            = "1-3"
                            frozen_headers        = "3"
                            absolute_column_width = "15, 10, 50, 50"
                            );

title1 j=l "CRF page number consistency check (define.xml vs annotated CRF)";
title2;

proc print data=ChkDefCrfPage label noobs;
run;

ods tagsets.excelxp close;
ods listing;

%mend ChkDefCrfPage;
```

VALIDATION

Now that we have define.xml metadata and annotations within a SDTM annotated CRF available as SAS datasets, we have the power of SAS to perform the CRF page number consistency check. Note that in this macro we check only CRF page of variables in parent domains since annotations of supplemental domains may have a different style. And you can develop other checks if needed. The advantage of this approach is that you have the ability and flexibility to add/delete the checks based upon your validation needs.

VALIDATION RESULT

We use ODS TAGSETS.EXCELXP to produce output in Excel. The validation report of CRF page number inconsistencies between annotated CRF and define.xml was shown below.

Domain Abbreviation	SDTM Variable	CRF Page in Define.xml	CRF Page in Annotated CRF
DD	DDDTC		115
DD	DDTESTCD	115	115, 116
LB	LBTEST	32, 33, 34, 37	32, 33, 34, 37, 39, 41, 43, 45, 47, 49
LB	LBTESTCD	30, 32, 33, 34, 35, 36, 37, 55, 56	30, 32, 33, 34, 35, 36, 37, 39, 41, 43, 45, 47, 49, 55, 56
MH	MHENTPT	13	
PC	PCTEST	138, 139	
TU	TUTEST	119, 123, 124, 127, 128, 132	
XZ	XZTESTCD	54, 117	54, 117, 118

Figure 2. Validation Result

For instance, the first row indicates that a variable DDDTC is annotated on page 115 of blankcrf.pdf, however, the CRF page number of variable in define.xml is missing, and therefore the origin of variable in define.xml does not indicate that this variable came from page 115. The fifth row shows that define.xml defines the variable MHENTPT is annotated on page 13 of blankcrf.pdf, however, page 13 of the blankcrf.pdf is missing the annotation.

CONCLUSION

The macro described in this paper is a very useful tool which can be used to make sure a define.xml that is accurate. With this macro, a programmer should be able to identify CRF page number inconsistencies that could have been missed if we had just relied on some validation tools or manual checking and able to look into variable presence issue (i.e. variable is listed in the define.xml but not annotated in blankcrf.pdf). Additionally, if we replace the macro parameter VARLIST value with all SDTM variables then we can use this macro to check if variable is annotated in blankcrf.pdf but not listed in the define.xml.

REFERENCE

FDA CDER, 2011. “Common Data Standards Issue Document.” Available at http://www.fda.gov/downloads/Drugs/DevelopmentApprovalProcess/FormsSubmissionRequirements/ElectronicSubmissions/UCM254113.pdf

Prafulla Girase, Robert Agostinelli. “Automating Validation of Define.xml using SAS”. PharmaSUG 2013. Available at http://www.lexjansen.com/pharmasug/2013/AD/PharmaSUG-2013-AD19.pdf

Wendi L. Wright. 2010. “How to Create an XML Map with the XML Mapper”. NESUG 2010. Available at http://www.lexjansen.com/nesug/nesug10/ff/ff14.pdf

Joel Campbell, Ryan Wilkins. “Importing and Parsing Comments From a PDF Document With Help From Perl Regular Expressions”. PharmaSUG 2011. Available at http://www.pharmasug.org/proceedings/2011/CC/PharmaSUG-2011-CC22.pdf

ACKNOWLEDGMENTS

I would like to thank my colleagues Dmitry Kolosov and Lianbo Zhang for helpful discussion.

CONTACT INFORMATION

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

Name: Xianhua Zeng
Enterprise: PAREXEL International
Address: 9F & Unit A/B/C 10F, No.506, Shangcheng Road, Pudong District, Shanghai, China, 200120
Work Phone: +86 21-51118305
Fax: +86 21 61609196
E-mail: Allen.Zeng@PAREXLE.com
Web: http://www.parexel.com/

Name: Shenglin Zhang
Enterprise: PAREXEL International
Address: 9F & Unit A/B/C 10F, No.506, Shangcheng Road, Pudong District, Shanghai, China, 200120
Work Phone: +86 21-20505357
Fax: +86 21 61609196
E-mail: Shenglin.Zhang@PAREXLE.com
Web: http://www.parexel.com/