PharmaSUG 2015 – AD09

SDTM Annotations: Automation by implementing a standard process

Geo Joy, Novartis, Cambridge, MA
Andre Couturier, Novartis, East Hanover, NJ

ABSTRACT

Annotating a Blank Case Report Form -- a collection of unique CRF pages stored in a file named BlankCRF.pdf is an important part in submission activity and a lot of productive time is spent to manually annotate each page. Maintaining consistency across annotations, validating them against submission data set are all manual and error prone activities. This paper talks about an effective way to automate the entire SDTM annotation process by setting up an annotation database and by reusing the annotations done by earlier studies with the help of SAS® and PDF editing tool. This is particularly useful as majority of the standard CRF pages are reused in multiple trials and programmers can annotate those pages in almost no time.

INTRODUCTION

Standardization of CRF pages is the starting point for annotation automation. If pages are not reused across trials there is no possible efficiency gain as each unique CRF page needs to be annotated manually each time. CRF Standardization ensures that a large proportion of pages are reused across different studies. Following a standard approach in designing CRF pages is crucial to improve the efficiency and accuracy of the data collection process. This in turn will make the automation of the annotation process, the management of the annotation library and database much simpler.

The second key requirement for automation is setting standard annotation rules across the organization to increase the quality of annotations by improving the consistency across unique pages that may be annotated by different individuals. Annotation rules increases efficiency by reducing the time required to align annotations format, color and text between pages.

Unique SDTM annotated CRF are maintained in PDF format in a central library with versioning and change control process. Each CRF SDTM annotation attribute (CRF ID, text, color, and position) is extracted with the help of SAS® and stored in a SAS® data set “the annotation database”. This database is the primary input for annotating any number of study CRFs. Creating the annotation database is an investment that requires maintenance and a change control process but has the potential to reduce the annotation cycle time tremendously. Whether automation is right for you will depend amongst other things on the size of your organization and the number of trials performed each year.

Figure 1 provides a high level overview of the automated SDTM annotation process.

In this paper we will go over SDTM annotation rules, and more functionalities of the annotation tool such as; How to easily link the database information to a Study BlankCRF.pdf automatically, reusing annotations from other annotated CRFs, bookmarking, extracting CRF SDTM annotation domain, variable name information to a data set to help the validation of CRF against submission data sets and Define.xml. For more details on creating and maintaining an annotation database, we recommend reading the PharmaSug paper by Walter Hufford “CC01 - Automating Production of the blankcrf.pdf “.

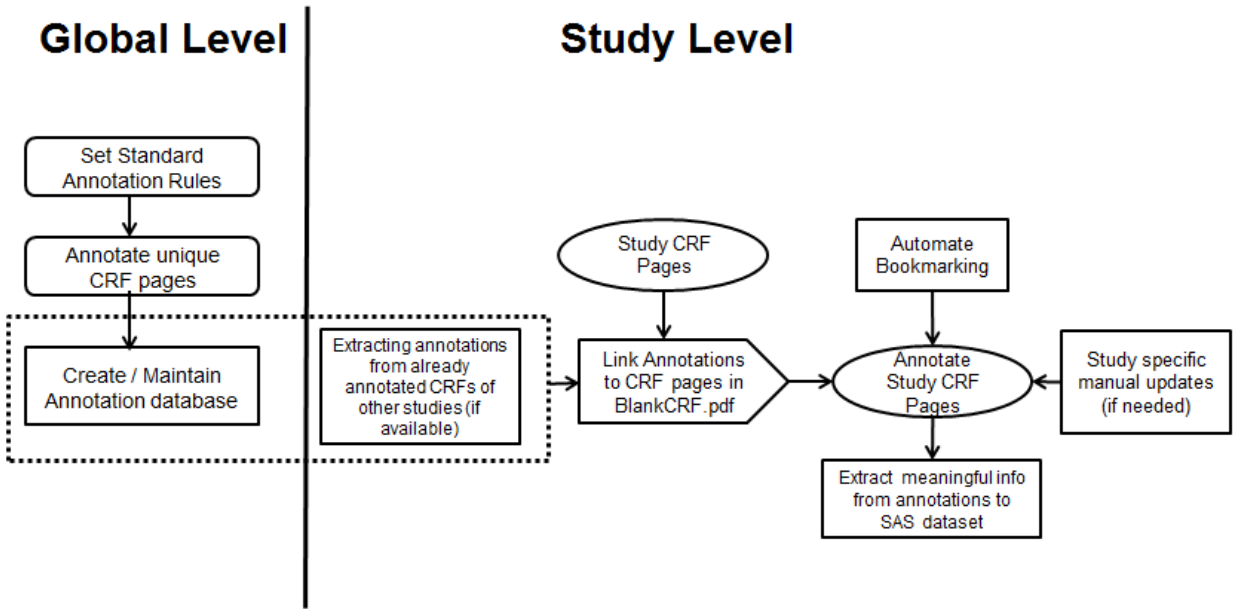

Figure 1: High level process flow of automation of SDTM Annotation.

DEFINING STANDARD ANNOTATION RULES

Because each unique CRF pages are annotated by different people, sometimes from different departments (Data Management or Statistical Reporting), SDTM annotation rules help ensure consistency when the different pages are assembled together for a specific trial. Defining standard annotation rules across an organization is vital in automating the annotation process.

There are actually different "right" ways of annotating a CRF page, say in the case of supplemental qualifiers, it can be annotated either as [SUPPXX.QVAL when QNAM =VARNAME] or as [VARNAME in SUPPXX]; or domain can be annotated as [DM = Demographics] and even as [Domain = DM].

Inconsistency in application of annotation rules can result in increased review time and limit the benefit of reusing annotations from a different study. i.e. SUPPQUAL on the first page could be annotated in the format "[SUPPXX.QVAL when QNAM =VARNAME] "whereas on the next page it could be annotated as "[VARNAME in SUPPXX]". Other annotation attributes such as font or color coding can also cause issues if they vary widely across pages.

Study Data Tabulation Model Metadata Submission Guidelines (SDTM-MSG), prepared by the CDISC SDS Metadata Team (Section 4) is a good reference paper to define standard annotation rules.

In Table 1, we provide a very common SDTM annotation construct patterns that we follow for our own SDTM annotations.

Annotation Pattern	**Example**
[Domain Short Name] = [Domain Long Name]	AE = Adverse Events
VARNAME	USUBJID AETERM AEDECOD etc.
VARNAME = VALUE	DSTERM = RANDOMIZED
VARNAME1/VARNAME2 = VALUE	DSTERM/DSDECOD = RANDOMIZED
VARNAME in SUPPXX	AEDIS in SUPPAE

VARNAME1/VARNAME2 in SUPPXX	ICDUR/ICDURU in SUPPZH
VARNAME1 when/where VARNAME2 = VALUE1	VSORRES when VSTESTCD = SYSBP
VARNAME1/VARNAME2 when/where VARNAME3 = VALUE1	VSORRES/VSORRESU when VSTESTCD = SYSBP
[NOT SUBMITTED] Entire Page: [NOT SUBMITTED]	<Don't extract anything. Skip>

Table 1: Common SDTM annotations construct patterns

ANNOTATION DATABASE

Annotation database is a SAS® data set maintained in a central location that stores SDTM annotation text and its attributes corresponding to each CRF page. This helps to reutilize these annotations to automatically annotate any number of studies in the future. Figure 2 shows a screenshot of the "Annotation Database" and the corresponding annotated CRF Page equivalent. To load annotations from any annotated CRF pages into the "Annotation Database, first export annotations to a FDF file using Adobe Acrobat and then extract attributes from the FDF file using a SAS program that searches for FDF specific lookup words.

The whole process of creating a database from a FDF file is explained in a very detailed way in the PharmaSUG paper "*CC01 - Automating Production of the blankcrf.pdf* ".

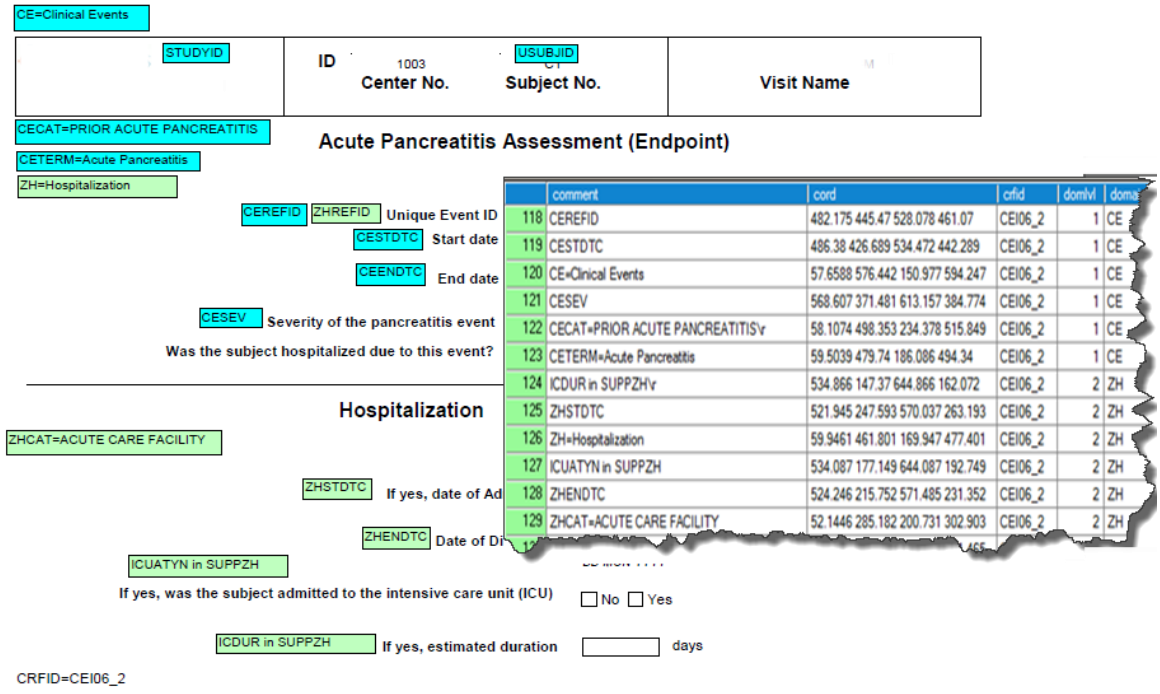

Figure 2: CRF Page annotations and the Annotation Database

EXTRACTING UNIQUE CRF PAGE IDENTIFIER FROM CRF PAGES

To reuse CRF annotations that were databased, we need a way to link the extracted annotations to the correct pages in Study BlankCRF.pdf. As shown in Figure 3, each page is identified with a unique CRF ID that is preceded with a keyword.

To automate the SDTM annotation process, the tool must read through the annotated CRF pages and extract page number and the corresponding CRF ID. However, since text in PDF Files is not directly readable using SAS® it should first be converted to a SAS® readable format such as ASCII/text file. There are quite a few open source command-line utilities / tools that can be invoked directly using "X command" in SAS®; we selected PDFTOTEXT that does a very good job in pdf to text conversion.

Figure 3 displays a code snippet to read text from a BlankCRF.pdf file and extract a CRF identifier using a keyword 'CRFID='

Figure 3: Code snippet from extract annotation module to extract CRF identifier.

REUSING ALREADY ANNOTATED CRF PAGES FROM SIMILAR STUDIES

Creating a central "Annotation Database" is a big step forward in SDTM Annotation automation process. But Sometimes, CRF annotations are missing from the database if it is very study specific and rarely used, or if it is newly created. Also in the case of CRF page variations (changes to a global page to satisfy study specific needs) annotations could be available in the database but a few may need some manual updates (adding, modifying or removing annotations). A simple review of the annotated BlankCRF file will highlight these issues and manual updates will be required to adapt the annotation for that study.

Since other studies in the same clinical project may benefit from these manually corrected and the newly annotated study pages, we added functionality based on the same process and logic for creating the central annotation database to extract annotations into "Study Annotation Database". A SAS® macro module (Extract Annotation module) was developed that automates that "Study Annotation Database" creation on the fly. Through the macro parameter FDFNAMES, users can supply as many input annotated BlankCRF.pdf (in the form of FDF files) from other studies, separated by a pipe sign, to be used for "Study Annotation Database" creation. Using this parameter, users can also prioritize annotation sources when the same CRF pages are available in multiple study sources based on the order provided.

Once extracted, this "Study Annotation Database" can also be used as input in the automation process to annotate other study CRF Pages.

Figure 4a shows a simple macro call to Extract Annotation module, Figure 4b shows a code snippet from Extract Annotation module to extract annotation attributes from FDF files using the KEYWORDs.

```
%extract_annotation(FDFNAMES=Study1.fdf|Study2.fdf, /*Annotated Study CRFs in priority
                                                      order*/
                    OUTDS   =work.Study_Database  );/*output data set with extracted
                                                      annotation attributes*/
```

Figure 4a: Macro call of Extract Annotation module to create Study Annotation Database.

```
/* Perl Expressions to extract each annotation attributes using the Keywords */
%let cordinate = %str(#/Rect\[(.*?)\]#io);    /* Keyword: /Rect[] to extract all the
                                                 annotation coordinates */
%let comment = %str(#Contents\((.*?)\)\/#io); /* Keyword: Contents()/ to extract
                                                 annotation text */
%let page = %str(#\/Page (.*?)\/#io);         /* Keyword: \Page \ to extract pages */
%let fill_color = %str(#\/C\[(.*?)\]#io);     /* Keyword: C[] to extract
                                                 fill in color */

%macro perl_attr_extract(Perl_expression = ,outvar=);
         rx = prxparse("&Perl_expression");
         start=1;
         stop=reclen;

         call prxnext(rx, start, stop, _infile_, position, length);
         if rx gt 0 and length gt 0 then do;;
           call prxposn(rx, 1, start, length);
           &outvar=substr(_infile_,start,length);
         end;
         else do;
           &outvar = ' ';
         end;
%mend perl_attr_extract;
```

Figure 4b: Perl expressions and code snippet within Extract Annotation Module to extract each annotation attributes from FDF files.

NEW STUDY ANNOTATION PROCESS

The study annotation process consist of creating a FDF file containing each BlankCRF.pdf page annotations feeding from the information that are either stored in the central “annotation database” or in “study annotation database” created from earlier studies (if available). Once built, the FDF file is imported in the BlankCRF.pdf file using Adobe Acrobat.

To build a study specific FDF file, the macro (study annotation module) first reads through the BlankCRF ASCII representation to get the CRF ID and corresponding page number (as shown in Figure 3). Using this unique CRF ID the macro then brings over all the matching annotations and attributes from the input sources. Now we have all the annotations, attributes along with the page number needed to annotate a Study BlankCRF.pdf at one place.
The macro then writes the FDF file using this info. If the same CRF page annotations are available in study and central annotation database user will have option to pick the preferred source.

FDF file always follow a consistent pattern; it will have a header in the beginning, then a body that corresponds to each annotations and a trailer in the end. Hence with a basic understanding of FDF file pattern, the study specific FDF file can easily be created using SAS® utilizing all these annotations and attributes. Figure 6 shows a sample FDF file and corresponding SAS® program used to create it. Adobe Acrobat can directly import annotations in any FDF file to a Study BlankCRF.pdf.

Figure 5 shows how using “Study Annotation database” could improve the automatic annotation rate.

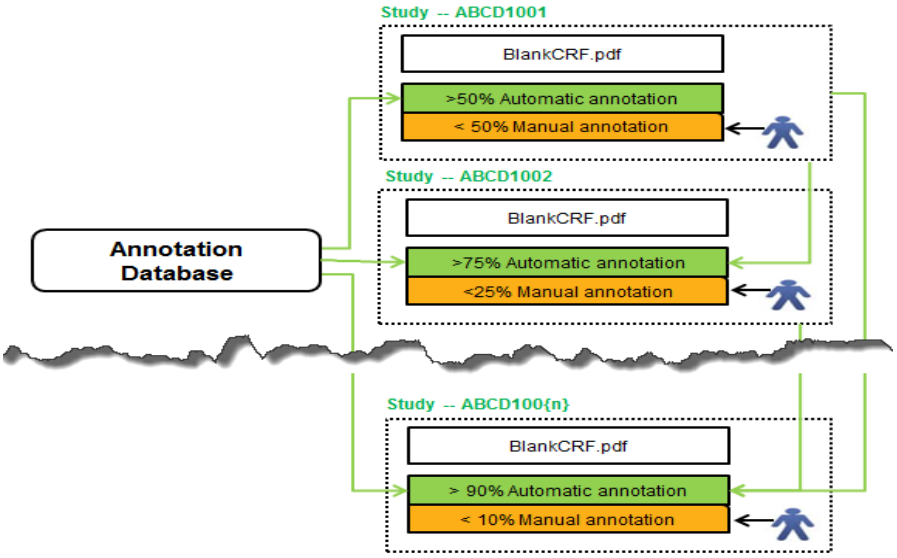

Figure 5: Annotation Database and re-use annotations from different studies

```
filename out "./study_fdf.fdf";

data _null_ ;
  file out ls=6000 ;
  set annot_final end=eof ;
  by PAGE;

      if _n_ =1 then do ;
        put  @1 '%FDF-1.2' /
           @1 _N_' 0 obj' /
            @1 '<< /FDF << /Annots [' %do i = 2 %to %eval(&sqlobs+1) ; "&i 0 R " %end ; '] >> >>' /
            @1 'endobj' ;
      end;

      N = _N_+1;

      put
        @1 N ' 0 obj' /
        @1 '<</BS 21 0 R/C[' fcol ']/Rotate 0/'
          'Contents(' comment ')/DA(0 0 0 rg /Helv 12 Tf)/'
          'DS(font: Arial,sans-serif 9.0pt; text-align:left; color:##000000 )/F 4/'
         'Page ' page '/Rect[' cord ']/Subj(TextBox)/Subtype/FreeText/T(NCDS)/Type/Annot>>' /
        @1 'endobj' ;
      if eof then do ;
       put
            @1 'trailer' /
            @1 '<< /Root 1 0 R >>' /
            @1 '%%EOF' ;
      end;
run;
```

```
1 0 obj
<</FDF<</Annots[2 0 R 3 0 R ]>> >>
endobj
2 0 obj
<</BS 33 0 R/C[0.0 1.0 1.0]/Rotate 0/Contents(STUDYID )/DA(0 0 0 rg /Helv 12 Tf)/
DS(font: Arial,sans-serif 9.0pt; text-align:left; color:##000000 )/F 4/Page 0/
/Rect[159.985 554.563 206.077 568.163]/Subj(TextBox)/Subtype/FreeText/T(NCDS)/Type/Annot>>
endobj
3 0 obj
<</BS 33 0 R/C[0.0 1.0 1.0]/Rotate 0/Contents(USUBJID )/DA(0 0 0 rg /Helv 12 Tf)/
DS(font: Arial,sans-serif 9.0pt; text-align:left; color:##000000 )/F 4/Page 0/
/Rect[401.697 554.17 445.183 567.77]/Subj(TextBox)/Subtype/FreeText/T(NCDS)/Type/Annot>>
endobj
trailer
<</Root 1 0 R>>
%%EOF
```

Header

Body

Trailer

Figure 6: SAS® program screenshot from "Study Annotation" macro module to create the above FDF file.

AUTOMATED BOOKMARKING

As per SDTM annotation Guidelines, bookmarking should be added to the annotated BlankCRF.pdf to provide reviewer the overview of the data collected for the study and help them navigate through the document. Bookmarking should be done in two ways (dual bookmarking)

- Bookmark CRF Topic by visit / time points in the study.
- Bookmark visit / time points by CRF Topic.

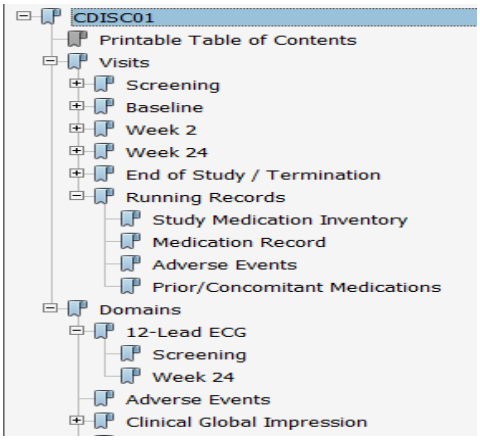

Figure 7: Bookmarks

Manual bookmarking is a long, tedious and error prone task, that needs to be repeated for every study. Making it an ideal candidate for automation, assuming we can get the required information. SAS® provides options to add bookmarking to pdf file that it creates by itself, but inserting bookmark to an already existing PDF file can be a bit tricky.

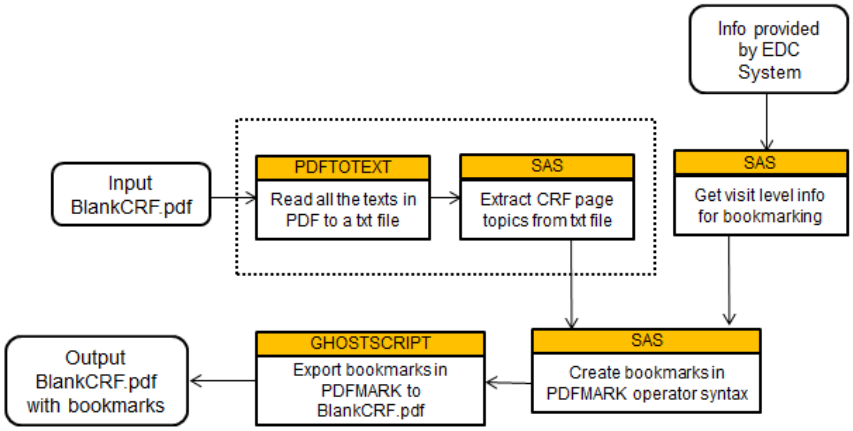

Figure 8: Process flow for automatic bookmarking.

Earlier section covers procedure to read any PDF file in SAS® by first converting it to a text file using PDFTOTEXT command line utility. Because of the way our CRF pages have been standardized, we can use the same technique to extract CRF topics from the converted text by using a specific lookup word. In our case the CRF topic always came after the text “Visit Name” in every CRF page.

The page number of each CRF page is also needed to link the bookmarks to the correct page. This can be done in several ways; using the knowledge that PDFTOTEXT utility puts a form feed characters as a page break by default, one could count those to get the page number. Or count any key text that uniquely appears on every CRF (such as ‘CRFID=’ used as a key text as shown in Figure 3).

Visit level info for each CRF page domain can usually be retrieved from the information provided by electronic data collection system.

To add these CRF topics and visit level information as linkable bookmarks to a Study BlankCRF.pdf, we use PDFMARK, a PostScript-language extension used by Acrobat Distiller to include features that are present in PDF, but not in standard PostScript. PDFMARK operator has a feature named OUT that allows adding any bookmarks with correct links to a page by following the PDFMARK syntax. Explaining the PDFMARK operator syntax, is beyond the scope of this paper, but you can find a lot of references by searching through the web. Figure 9 shows a good example of PDFMARK operator, its syntax and the bookmarks it generated. If you carefully examine the PDFMARK screenshot you can see that the PDFMARK file is something that can be created on the fly using SAS® with the extracted CRF topics, visit and the corresponding page info. The final step is to import the generated bookmarks in PDFMARK to the Study BlankCRF.pdf. For that we utilized an external open source software called Ghostscript and invoked it through SAS® X command using the 'GS' command.

Figure 10 shows SAS® code that exports bookmarks from pdfmarks file stored in the home directory to the original "BlankCRF.pdf" file using Ghostscript 'GS' and SAS® 'X' command.

```
%% parent bookmark (Domain) that has 2 children (Demography and Vital Signs)
[/Count 2
/Page 1
/View [ /Fit ]
/Title (Domain)
/OUT pdfmark
%%Demography bookmark with one child(Screen)
[/Count 1
/Page 2
/View [ /Fit ]
/Title (Demography)
/OUT pdfmark
%%First Child
[/Count 0
/Page 3
/Title (Screen)
/OUT pdfmark
%%Vital SIgns bookmark with three child(Screen Baseline Week24)
[/Count 3
/Page 3
/View [ /Fit ]
/Title (Vital Signs)
/OUT pdfmark
%%First Child
[/Count 0
/Page 3
/Title (Screen)
/OUT pdfmark
%%Second Child
[/Count 0
/Page 3
/Title (Baseline)
/OUT pdfmark
%%Third Child
[/Count 0
/Page 3
/Title (Week 24)
/OUT pdfmark
```

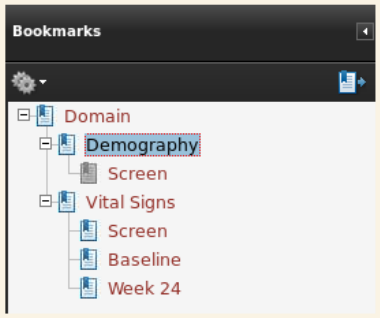

Figure 9: PDFMARK operator example with the output bookmarking screenshot.

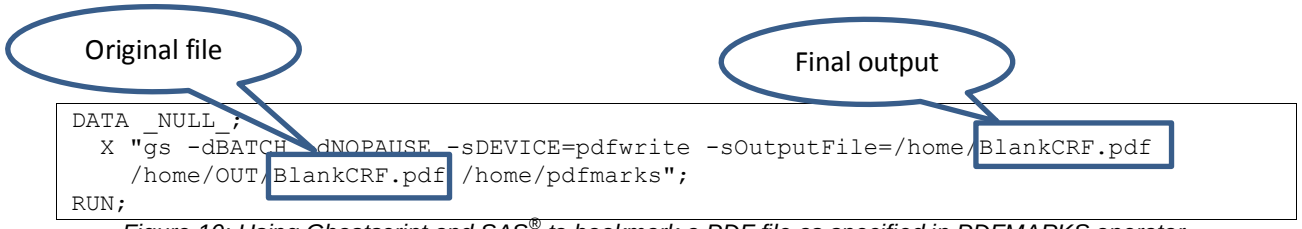

Figure 10: Using Ghostscript and SAS® to bookmark a PDF file as specified in PDFMARKS operator.

EXTRACTING DOMAIN AND VARIABLE NAMES FROM ANNOTATIONS

Define.xml requires hyperlinks to the annotated BlankCRF.pdf. Extracting page number, domain and variable name automatically from annotated BlankCRF.pdf is useful to build these hyperlinks directly to define.xml. Along with increasing efficiency this process also greatly reduces the margin for error. This information can also be used as an aid to validate SDTM annotations by comparing them against the submission data or vice-versa. E.g. identifying collected data missing from SDTM data sets.

comment	domain	varname	page
PHCAT = LIVER PATHOLOGY	PH	PHCAT	1
PHDTC	PH	PHDTC	1
PHMETHOD = LIVER BIOPSY	PH	PHMETH...	1
PHORRES when PHTESTCD=OINTP	PH	PHORRES	1
PHORRES when PHTESTCD=OINTP	PH	PHTESTCD	1
STUDYID	PH	STUDYID	1
USUBJID	PH	USUBJID	1

Figure 11: Screenshot of SAS® data set with domain name and variable name extracted from the annotations.

Once a Study BlankCRF.pdf is annotated, the “Extract Annotation” module should be run to extract the annotation text, corresponding page numbers, fill in color and other annotation attributes for each annotation into a SAS® data set. Domain annotation can be identified from the annotation texts by looking at the annotations in the format [Domain Short Name] = [Domain Long Name]. One of the biggest challenges is identifying annotations that correspond to a particular domain especially when multiple domains are annotated in a single CRF page. Setting a specific standard annotation rule across the organization opens a way to achieve this. With the knowledge that fill in color of each annotation can be extracted from FDF file by searching the look up text “/C[color code]/”, this problem can easily be resolved if we set the rule to use the same fill color for domain and corresponding annotations.

To extract variable name from each annotations, we created Perl regular expressions for each defined annotation patterns (as shown in Table 2) and then identified the number of variables that need to be extracted from each annotation text by matching it with its Perl expression and looping it through each annotation to extract the corresponding variables. Figure 12 shows a piece of code that does this job in one go.

Annotation Pattern	**Perl Expressions**
VARNAME	/^\w{3,8}$/i
VARNAME = VALUE	/^\w{3,8} ?= ?\S+/i
VARNAME1/VARNAME2 = VALUE	/^\w{3,8}\/\w{3,8} ?= ?\S+/i
VARNAME in SUPPXX	/^\w{3,8} IN SUPP\w{3,4}$/i
VARNAME1/VARNAME2 in SUPPXX	/^\w{3,8}\/\w{3,8} IN SUPP\w{2,4}$/i
VARNAME1 when/where VARNAME2 = VALUE1	/^\w{3,8} WHE(N|RE) \w{3,8} ?= ?\w+/i
VARNAME1/VARNAME2 when/where VARNAME3 = VALUE1	/^\w{3,8}\/\w{3,8} WHE(N|RE) \w{3,8} ?= ?\S+/i

Table 2: Perl Expressions and SDTM annotations construct patterns

```
DATA _ext_dom_var_temp1;
 /* the first column below defines patterns using Perl Regular Expressions;
    the second column specifies the number of variables (VARNAME) to be extracted*/
      array annopat(7,2)$100 _TEMPORARY_ (
            '/^\w{3,8}$/i'                                    , '1'
            '/^\w{3,8} IN SUPP\w{2,4}$/i'                     , '1'
            '/^\w{3,8} ?= ?\S+/i'                             , '1'
            '/^\w{3,8}\/\w{3,8} IN SUPP\w{2,4}$/i'            , '2'
            '/^\w{3,8}\/\w{3,8} ?= ?\S+/i'                    , '2'
            '/^\w{3,8} WHE(N|RE) \w{3,8} ?= ?\w+/i'           , '2'
            '/^\w{3,8}\/\w{3,8} WHE(N|RE) \w{3,8} ?= ?\S+/i' , '3'
      );
      SET _ext_dom_var_temp(rename=(comment=_comm));

/* get the corresponding group code when annotation matches the defined pattern */
         DO i=1 TO dim(annopat,1);
          if prxmatch(annopat(i,1),strip(_comm)) gt 0 then _anotyp=annopat(i,2);
         END;
         _numid=input(_anotyp,best.);

/* remove the word WHERE or WHEN before extraction of VARNAME */
         _comm=prxchange('s/WHE(N|RE)//',1,_comm);

/* extract VARNAMEs separated by the expected characters */
         DO j=1 TO _numid;
            varname=strip(scan(_comm,j,' =/'));
            OUTPUT;
         END;
RUN;
```

Figure 12: Code snippet to extract variable name from annotations using Perl expressions.

CONCLUSION

In this paper we showed that with standard annotation rules, some knowledge of PDF, and FDF file structure, a few Open Source utilities / tools, automation of the annotation process is possible. This process can help automate the entire process from annotation, bookmarking and some validation of define.xml and submission data sets. The resulting annotated BlankCRF.pdf still requires a thorough review by the trial programmer but the gain in efficiency is phenomenal. What used to take a week or more can now be completed in a day or even hours when the previous trials in the same project have already been annotated.

Our next step is to build an interface and incorporate SAS® stored process so that data managers can deliver annotated CRF to our vendors and programming group immediately after the database is finalized and without intervention from the programming group.

REFERENCES

Hufford, Walter. “Automating Production of the blankcrf”
Available at http://www.pharmasug.org/proceedings/2014/CC/PharmaSUG-2014-CC21.pdf.

Study Data Tabulation Model Metadata Submission Guidelines (SDTM-MSG), prepared by the CDISC SDS Metadata Team
Available at http://www.cdisc.org/stuff/contentmgr/files/0/4254da3fdc854804b27881ca82d24051/misc/metadata.zip

PDFMARK Reference Manual
Available at http://www.adobe.com/content/dam/Adobe/en/devnet/acrobat/pdfs/pdfmark_reference.pdf

ACKNOWLEDGMENTS

We would like to thank Xiaogang Luan and Lily Peng for their review and suggestions.

CONTACT INFORMATION

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

Geo Joy
Novartis Institutes for BioMedical Research, Inc.
45 Sidney Street
Cambridge, MA 02139
USA
geo.joy@novartis.com
www.novartis.com

Andre Couturier
Novartis Pharmaceuticals Corporation
One Health Plaza
East Hanover, NJ 07936-1080
USA
andre.couturier@novartis.com
www.novartis.com