

PharmaSUG 2020 – Paper AD-208

Programming Patient Narratives Using Microsoft Word XML files

Yuping Wu, PRA Health Science

Jan Skowronski, Genmab A/S

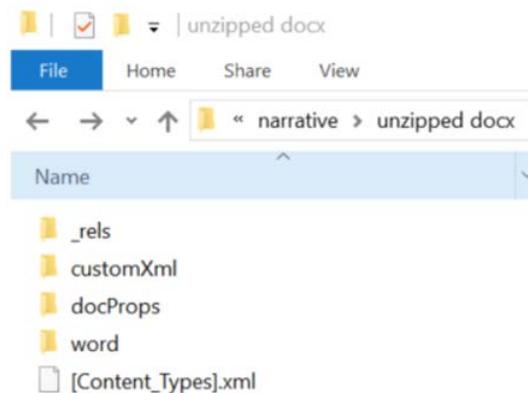
ABSTRACT

Patient narratives are an important component of a clinical study report (CSR). The ICH-E3 guidelines require a brief narrative to describe each death, each serious adverse event and other significant adverse events that are judged to be of special interest because of clinical importance. Different from individual tables and listings in a CSR, patient profiles present most, if not all, of the information collected for a subject over the course of a clinical trial. Such documents typically integrate the various sources of data into one cohesive output with each part present in different formats which pose challenging problems for both programmers and medical writers. This paper introduces a new method that uses SAS together with Office Open XML. The narrative Word templates layout is created separately with properly tagged cells. The Word file is then broken up into its XML parts. Read and updated by SAS with the clinical data and finally saved back into its original format which now contains a populated narrative.

INTRODUCTION

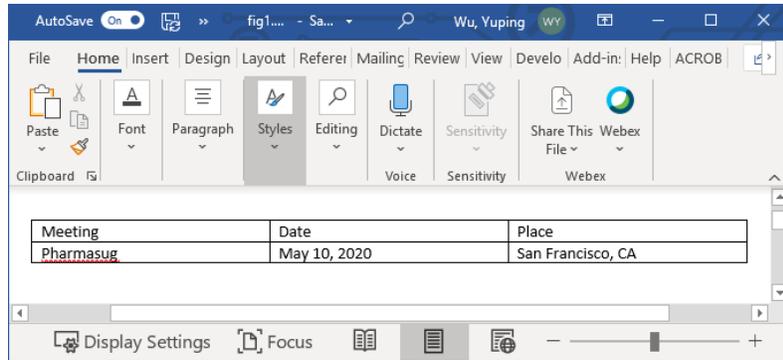
Patient narratives are a part of safety data submitted to regulatory authorities. As per ICH E3 guidelines [1] a narrative is needed to describe serious adverse events including death and other significant adverse events that are of special interest for the study. The detail descriptions include the nature and intensity of event, the clinical course leading up to event, relevant laboratory measurements, drug administration detail and its relationship to the clinical events and investigator's opinion on causality. These guidelines also state that the narrative should include patient identifier, age and sex of patient, disease being treated, relevant illnesses, relevant medications and drug administration details. All these various types of data should be presented in one narrative output. Not like tables and listings that have consistent format narratives typically contain multiple different formats in one output because of the nature of data organization which makes it unfriendly for the common used output procedures such as proc report and proc print. The goal of this paper is to introduce a new method that use SAS together with Office Open XML. With this method the structure of the narrative Word template is preserved in output while the tagged cells are updated by clinical data, therefore making it fit for any type of data organization and presentation formats.

Word DOCX file is a ZIP archive of XML files [2]. When unzipping any word DOCX document you can see the file structure shown in Display 1. All these files are XML files and each of them are part of content of a word document. Among them DOCUMENT.XML under word folder is the main XML that contains the text information including text, text format, style, font etc., which we will elaborate in detail in an example. Shown in Display 2 are a simple 2X3 table and its DOCUMENT.XML. We can see that XML elements have a hierarchical structure. For this table, texts are embedded in **w:t** elements within **w:r** elements within **w:p** elements within cell elements, **w:tc** within



Display 1. File Structure of Unzipped DOCX

row elements, **w:tr** within table elements, **w:tbl** within body elements, **w:body** within **w:document**. For this example of a 2x3 table, there are one table element, **w:tbl**, two row elements, **w:tr** under table element and three cell elements **w:tc** under each row element. The ability of SAS to read XML files offers a convenient method to extract structure information of a word based template and to edit the text and structure information of the template in SAS, which can be especially useful for automated generation of a large scale of patient narratives.



```

<w:document mc:Ignorable="w14 w15 w16se w16cid wp14" xmlns:wps="http://schemas.microsoft.com/office/word/2010/word"
xmlns:wpi="http://schemas.microsoft.com/office/word/2010/wordprocessingink" xmlns:wpg="http://schemas.microsoft.com
xmlns:w16se="http://schemas.microsoft.com/office/word/2015/wordml/symx" xmlns:w16cid="http://schemas.microsoft.com
xmlns:w14="http://schemas.microsoft.com/office/word/2010/wordml" xmlns:w="http://schemas.openxmlformats.org/wordp
xmlns:wp="http://schemas.openxmlformats.org/drawingml/2006/wordprocessingdrawing" xmlns:wp14="http://schemas.mic
xmlns:m="http://schemas.openxmlformats.org/officeDocument/2006/math" xmlns:r="http://schemas.openxmlformats.org/o
xmlns:am3d="http://schemas.microsoft.com/office/drawing/2017/model3d" xmlns:sink="http://schemas.microsoft.com/office
xmlns:cx1="http://schemas.microsoft.com/office/drawing/2016/5/19/chartx" xmlns:cx2="http://schemas.microsoft.com/ol
xmlns:cx5="http://schemas.microsoft.com/office/drawing/2016/5/11/chartx" xmlns:cx4="http://schemas.microsoft.com/ol
xmlns:cx2="http://schemas.microsoft.com/office/drawing/2015/10/21/chartx" xmlns:cx1="http://schemas.microsoft.com/ol
xmlns:wpc="http://schemas.microsoft.com/office/word/2010/wordprocessingCanvas">
  <w:body>
    <w:tbl>
      <w:tblPr>
        <w:tblGrid>
          <w:tblRow w:rsidR="002A3220" w14:tblRowID="77777777" w14:paraId="1DB32085" w:rsidR="002A3220">
            <w:tc>
              <w:tblPr>
                <w:p w14:tblRowID="3255332F" w14:paraId="5799B12C" w:rsidR="002A3220" w:rsidRDefault="00944C86">
                  <w:tr>
                    <w:td>Meeting</w:td>
                  </w:tr>
                </w:tblPr>
              </w:tc>
            <w:tc>
              <w:tblPr>
                <w:p w14:tblRowID="3255332F" w14:paraId="5799B12C" w:rsidR="002A3220" w:rsidRDefault="00944C86">
                  <w:tr>
                    <w:td>Date</w:td>
                  </w:tr>
                </w:tblPr>
              </w:tc>
            <w:tc>
              <w:tblPr>
                <w:p w14:tblRowID="3255332F" w14:paraId="5799B12C" w:rsidR="002A3220" w:rsidRDefault="00944C86">
                  <w:tr>
                    <w:td>Place</w:td>
                  </w:tr>
                </w:tblPr>
              </w:tc>
            </w:tblRow>
          <w:tblRow w:rsidR="002A3220" w14:tblRowID="77777777" w14:paraId="29289599" w:rsidR="002A3220">
            <w:tc>
              <w:tblPr>
                <w:p w14:tblRowID="3255332F" w14:paraId="5799B12C" w:rsidR="002A3220" w:rsidRDefault="00944C86">
                  <w:tr>
                    <w:td>Pharmasug</w:td>
                  </w:tr>
                </w:tblPr>
              </w:tc>
            <w:tc>
              <w:tblPr>
                <w:p w14:tblRowID="3255332F" w14:paraId="5799B12C" w:rsidR="002A3220" w:rsidRDefault="00944C86">
                  <w:tr>
                    <w:td>May 10, 2020</w:td>
                  </w:tr>
                </w:tblPr>
              </w:tc>
            <w:tc>
              <w:tblPr>
                <w:p w14:tblRowID="3255332F" w14:paraId="5799B12C" w:rsidR="002A3220" w:rsidRDefault="00944C86">
                  <w:tr>
                    <w:td>San Francisco, CA</w:td>
                  </w:tr>
                </w:tblPr>
              </w:tc>
            </w:tblRow>
        </w:tblGrid>
      </w:tblPr>
    </w:tbl>
  </w:body>
</w:document>

```

PROCESS OF PROGRAMMING PATIENT NARRATIVES WITH NEW METHOD

Several methods have been developed for automated generation of patient narratives [3]. All these methods used the same principles as we programmed tables and listings. They start with narrative shell and then create a data set where variables correspond to the columns in the shell and then use proc report or proc print procedure to generate output, adjust format and add titles and footnotes. This is very efficient for an output with many repeated rows but is less efficient for creating narratives where most of information in each row or cell is just presented once. Figure 1 illustrates the flow of narrative generation with our new method. The new method requires a clean and output-like template with all title, footnotes and headers in place. The narrative template is a format for data organization and presentation. It is typically created by medical writers and biostatistician. Programmers may update it to make sure all cells and rows fit the data for all patients. Secondly, work with medical writer to develop the specifications for the template and the format of presentation. Thirdly, tag each cells and rows with unique codes. It is important to start the codes with the same special symbol. In our example we use '#', but it can be others like '@' or '\$'. This symbol will be used as an indication where the clinical data should be placed in DOCUMENT.XML. Therefore, we need to make sure that this symbol is not found in other places of DOCUMENT.XML and the whole template. For the repeating rows we suggest using continuous numbers as codes to facilitate dynamical programming for

Display 2. A Word Table Example and its DOCUMENT.XML Layout

data set construction and row addition process. No cells or rows can have the same codes because they represent their clinical data and where the data are placed in DOCUMENT.XML. Fourthly, unzip the coded word template to XML files. DOCX files are XML archived documents. To unzip, just rename .DOCX to .ZIP and then follow the common unzipping procedure. It will be broken to many XML files. Among them DOCUMENT.XML contains text and structure information of the word file. We will read it into SAS for editing and updating. Fifthly, construct SAS data set. SAS data set used in new method is very different from the SAS data set used in traditional programming process. It just needs three variables, one is for subject ID, one for codes used in template, the other is for the corresponding clinical data of each unique code. The code value in the data set should be matched with the clinical data specified in narrative specifications and the codes in the template will be replaced by their clinical data in SAS data set in narrative output. For example, USUBJID=ABC-001001, CODE=#USUBJID, DESCRPT=001001, the cell with code=#USUBJID in template will be replaced by 0001001 in output. Constructing such data set can be easy for some of codes when the corresponding value can be directly obtained from SDTMs or ADaMs, but it can also get very complex when multiple pieces of information are needed to be integrated together in one record such as clinical event part. Sixthly, update DOCUMENT.XML with clinical data. read DOCUMENT.XML into SAS along with narrative data set which will be nested in hash table. when the unique code in DOCUMENT.XML are identified and matched with code in narrative data set it will be updated by the corresponding clinical data in narrative data set. If there is not corresponding clinical data code will be removed from DOCUMENT.XML. Therefore, the updated DOCUMENT.XML will keep all the structure information created in template but have the unique codes replaced by their corresponding clinical data. Seventhly, zip updated XML files into word output. Word document can be broken down to individual XML files. Zipping all XML files together also creates word file [4]. Since the updated DOCUMENT.XML is used the new word file will have their unique codes in template replaced by clinical data while all other formats remain the same as the original template. Finally, the programmed narratives are delivered to medical writers who will review and edit them with some non-CRF data. We will use two examples to further illustrate the whole process.

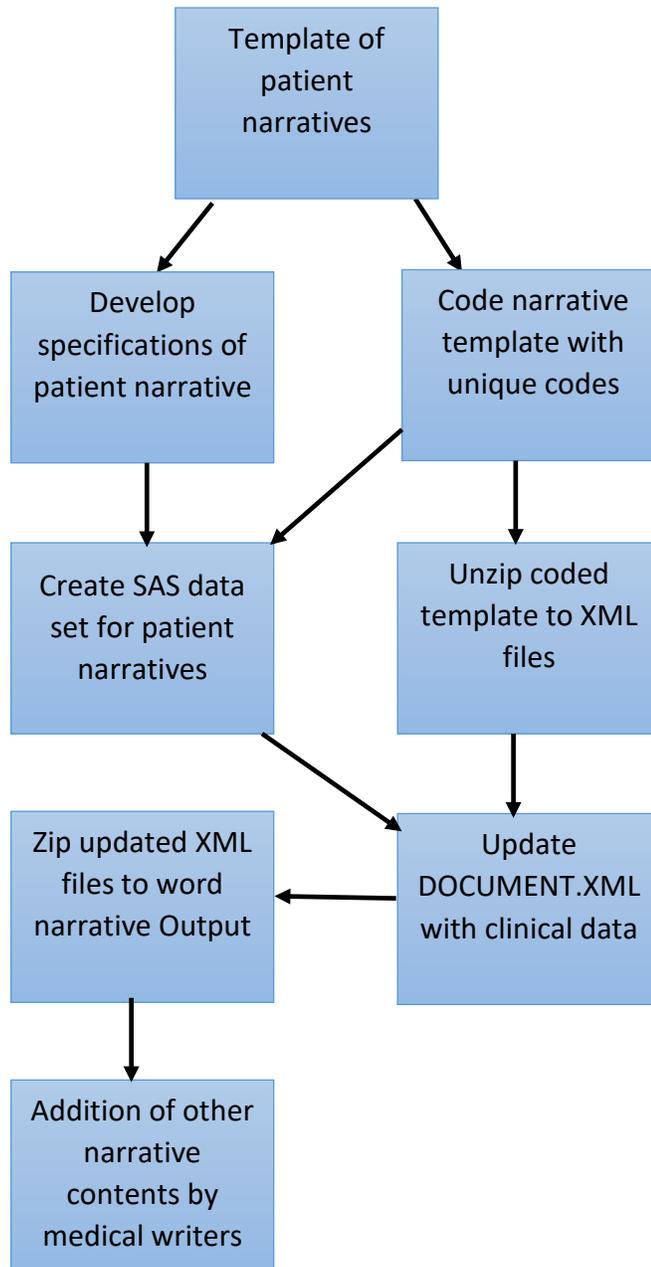


Figure 1, Process of Narrative Generation

will be updated by the corresponding clinical data in narrative data set. If there is not corresponding clinical data code will be removed from DOCUMENT.XML. Therefore, the updated DOCUMENT.XML will keep all the structure information created in template but have the unique codes replaced by their corresponding clinical data. Seventhly, zip updated XML files into word output. Word document can be broken down to individual XML files. Zipping all XML files together also creates word file [4]. Since the updated DOCUMENT.XML is used the new word file will have their unique codes in template replaced by clinical data while all other formats remain the same as the original template. Finally, the programmed narratives are delivered to medical writers who will review and edit them with some non-CRF data. We will use two examples to further illustrate the whole process.

EXAMPLE 1

Display 3 below shows the process of narrative generation for a dummy subject. The example of template includes an irregular table to collect subject ID and treatment information and a close table to list concomitant medications. This type of shell is not friendly to use proc report or proc print procedures,

Subject Number		Trial Number		
Trial Clinical Indication				
Trial Treatment				
Relevant Concomitant Medications				

Narrative Template

Subject Number	Subject number	Trial Number	Trial number	[Arm]
Trial Clinical Indication	Cancer type			
Trial Treatment	Drug A		[EXROUT] [EXDOSE] [EXDOSU] [EXDOSFRQ]	
Relevant Concomitant Medications				
[CMDECOD] [CMROUTE] [CMDOSE] [CMDOSU], from <<Study Day of CM start>> to <<Study Day of CM end>>; for [CMINDC]				
E.g. LORATADINE ORAL 10 mg, from Day -4 to Day 30; for insomnia PANTOPRAZOLE ORAL 40 mg, from Day 10 to ongoing; for gastric protection Rule for imputation of study day If less than or equal two years then use day; if more than two years then approximate to year, month.				

Specifications

Subject Number	#USUBJID	Trial Number	#STUDYID	#PHPACO
Trial Clinical Indication	#INDICAT			
Trial Treatment	Drug A		#RODOFR	
Relevant Concomitant Medications				
#CM001				

Coded Narrative Template

USUBJID	CODE	DESCRPT
ABC-001001	#INDICAT	CANCER
ABC-001001	#PHPACO	Phase 1 / Escalation Part / 50000mg
ABC-001001	#RODOFR	Oral 50000mg 10Q4W
ABC-001001	#STUDYID	ABC001
ABC-001001	#USUBJID	001001
ABC-001001	#CM001	CM1 ORAL 4000 mg, from Day -200 to Day 2; for AE1</w:t><w:br/><w:t>CM2 ORAL 20mg QID, from Day -10 to Day 21; for AE2</w:t><w:br/><w:t>CM3 ORAL 40 mg, from Day -6 to Ongoing; for AE3</w:t><w:br/><w:t>CM4 ORAL 2g BID, from Day 2 to Ongoing; for AE4</w:t><w:br/><w:t>CM5 ORAL 300mg TID, from Day 2 to Ongoing; for AE5</w:t><w:br/><w:t>CM6 ORAL 500mg BID, from Day 5 to Day 10; for AE6

Narrative SAS Data set

```

<?xml version="1.0" encoding="UTF-8" standalone="true"?>
<w:document mc:Ignorable="w14 w15 w16se w16cid wp14" xmlns:wps="http://schemas.microsoft.com/office/word/2010/wordprocessingSI"
xmlns:wpi="http://schemas.microsoft.com/office/word/2010/wordprocessingInk" xmlns:wpg="http://schemas.microsoft.com/office/word/
xmlns:w16se="http://schemas.microsoft.com/office/word/2015/wordml/symex" xmlns:w16cid="http://schemas.microsoft.com/office/word/
xmlns:w14="http://schemas.microsoft.com/office/word/2010/wordml" xmlns:w="http://schemas.openxmlformats.org/wordprocessingml/
xmlns:wp="http://schemas.openxmlformats.org/drawingml/2006/wordprocessingDrawing" xmlns:wp14="http://schemas.microsoft.com/office/word/2010/wordml"
xmlns:m="http://schemas.openxmlformats.org/officeDocument/2006/math" xmlns:r="http://schemas.openxmlformats.org/officeDocument/2006/relationships"
xmlns:am3d="http://schemas.microsoft.com/office/drawing/2017/model3d" xmlns:a:ink="http://schemas.microsoft.com/office/drawing/2010/chartex"
xmlns:cx8="http://schemas.microsoft.com/office/drawing/2016/5/14/chartex" xmlns:cx7="http://schemas.microsoft.com/office/drawing/2016/5/11/chartex"
xmlns:cx5="http://schemas.microsoft.com/office/drawing/2016/5/11/chartex" xmlns:cx4="http://schemas.microsoft.com/office/drawing/2015/10/21/chartex"
xmlns:cx2="http://schemas.microsoft.com/office/drawing/2015/10/21/chartex" xmlns:cx1="http://schemas.microsoft.com/office/drawing/2010/chartex"
xmlns:wpc="http://schemas.microsoft.com/office/word/2010/wordprocessingCanvas">
  <w:body>
    <w:tbl>
      <w:tblPr>
        <w:tblGrid>
          <w:tr w:rsidTr="00F51C48" w14:textId="77777777" w14:paraId="477DF22A" w:rsidRPr="00AC11C6" w:rsidR="00E1163E">
            <w:tr w:rsidTr="00F51C48" w14:textId="77777777" w14:paraId="3BCDEC4E" w:rsidRPr="00AC11C6" w:rsidR="00E1163E">
              <w:tr w:rsidTr="00F51C48" w14:textId="77777777" w14:paraId="4781C34F" w:rsidRPr="00AC11C6" w:rsidR="00E1163E">
                </w:tbl>
              <w:p w14:textId="2FC73206" w14:paraId="006DB53E" w:rsidR="00E1163E" w:rsidRDefault="00E1163E"/>
            </w:tbl>
          <w:tblPr>
            <w:tblGrid>
              <w:tr w:rsidTr="00F51C48" w14:textId="77777777" w14:paraId="60200DF3" w:rsidRPr="00AC11C6" w:rsidR="00E1163E">
                <w:tr w:rsidTr="00F51C48" w14:textId="77777777" w14:paraId="248CBA25" w:rsidRPr="00AC11C6" w:rsidR="00E1163E">
                  <w:trPr>
                    <w:tc>
                      <w:tcPr>
                        <w:p w14:textId="77777777" w14:paraId="48C40F53" w:rsidR="00E1163E" w:rsidP="00F51C48" w:rsidRDefault="00E1163E">
                          <w:pPr>
                            <w:r>
                              <w:rPr>
                                <w:t>#CM001</w:t>
                              </w:r>
                            </w:r>
                          </w:p>
                        </w:tc>
                      </w:tr>
                    </w:tbl>
                  <w:p w14:textId="07425D78" w14:paraId="07D6E2E7" w:rsidR="00E1163E" w:rsidRDefault="00E1163E"/>
                  <w:p w14:textId="1ACD3B5D" w14:paraId="18088B1B" w:rsidR="001D0A8A" w:rsidRDefault="001D0A8A"/>
                  <w:p w14:textId="6B7847C6" w14:paraId="4E6B2D3C" w:rsidR="001D0A8A" w:rsidRDefault="001D0A8A"/>
                  <w:p w14:textId="7126DC72" w14:paraId="00665711" w:rsidR="00265DC1" w:rsidRDefault="00265DC1"/>
                  <w:p w14:textId="77777777" w14:paraId="7AE4073C" w:rsidR="00265DC1" w:rsidRDefault="00265DC1">
                    <w:sectPr w:rsidR="00265DC1">
                      </w:body>
                    </w:document>

```

DOCUMENT.XML of Coded Template

```

<?xml version="1.0" encoding="UTF-8" standalone="true"?>
<w:document mc:Ignorable="w14 w15 w16se w16cid wp14" xmlns:wps="http://schemas.microsoft.com/office/word/2010/wordprocessingSI"
xmlns:wpi="http://schemas.microsoft.com/office/word/2010/wordprocessingInk" xmlns:wpg="http://schemas.microsoft.com/office/word/
xmlns:w16se="http://schemas.microsoft.com/office/word/2015/wordml/symex" xmlns:w16cid="http://schemas.microsoft.com/office/word/
xmlns:w14="http://schemas.microsoft.com/office/word/2010/wordml" xmlns:w="http://schemas.openxmlformats.org/wordprocessingml/
xmlns:wp="http://schemas.openxmlformats.org/drawingml/2006/wordprocessingDrawing" xmlns:wp14="http://schemas.microsoft.com/office/word/2010/wordml"
xmlns:m="http://schemas.openxmlformats.org/officeDocument/2006/math" xmlns:r="http://schemas.openxmlformats.org/officeDocument/2006/relationships"
xmlns:am3d="http://schemas.microsoft.com/office/drawing/2017/model3d" xmlns:a:ink="http://schemas.microsoft.com/office/drawing/2010/chartex"
xmlns:cx8="http://schemas.microsoft.com/office/drawing/2016/5/14/chartex" xmlns:cx7="http://schemas.microsoft.com/office/drawing/2016/5/11/chartex"
xmlns:cx5="http://schemas.microsoft.com/office/drawing/2016/5/11/chartex" xmlns:cx4="http://schemas.microsoft.com/office/drawing/2015/10/21/chartex"
xmlns:cx2="http://schemas.microsoft.com/office/drawing/2015/10/21/chartex" xmlns:cx1="http://schemas.microsoft.com/office/drawing/2010/chartex"
xmlns:wpc="http://schemas.microsoft.com/office/word/2010/wordprocessingCanvas">
  <w:body>
    <w:tbl>
      <w:p w14:textId="2FC73206" w14:paraId="006DB53E" w:rsidR="00E1163E" w:rsidRDefault="00E1163E"/>
    </w:tbl>
    <w:tbl>
      <w:tblPr>
        <w:tblGrid>
          <w:tr w:rsidTr="00F51C48" w14:textId="77777777" w14:paraId="60200DF3" w:rsidRPr="00AC11C6" w:rsidR="00E1163E">
            <w:tr w:rsidTr="00F51C48" w14:textId="77777777" w14:paraId="248CBA25" w:rsidRPr="00AC11C6" w:rsidR="00E1163E">
              <w:trPr>
                <w:tc>
                  <w:tcPr>
                    <w:p w14:textId="77777777" w14:paraId="48C40F53" w:rsidR="00E1163E" w:rsidP="00F51C48" w:rsidRDefault="00E1163E">
                      <w:pPr>
                        <w:r>
                          <w:rPr>
                            <w:rFonts w:cs="Times New Roman" w:hAnsi="Times New Roman" w:ascii="Times New Roman"/>
                            <w:sz w:val="20"/>
                            <w:szCs w:val="20"/>
                          </w:r>
                        </w:p>
                      </w:tc>
                    </w:tr>
                  </w:tbl>
                <w:p w14:textId="07425D78" w14:paraId="07D6E2E7" w:rsidR="00E1163E" w:rsidRDefault="00E1163E"/>
                <w:p w14:textId="1ACD3B5D" w14:paraId="18088B1B" w:rsidR="001D0A8A" w:rsidRDefault="001D0A8A"/>
                <w:p w14:textId="6B7847C6" w14:paraId="4E6B2D3C" w:rsidR="001D0A8A" w:rsidRDefault="001D0A8A"/>
                <w:p w14:textId="7126DC72" w14:paraId="00665711" w:rsidR="00265DC1" w:rsidRDefault="00265DC1"/>
                <w:p w14:textId="77777777" w14:paraId="7AE4073C" w:rsidR="00265DC1" w:rsidRDefault="00265DC1">
                  <w:sectPr w:rsidR="00265DC1">
                    </w:body>
                  </w:document>

```

Updated DOCUMENT.XML of Narrative Template

Subject Number	001001	Trial Number	ABC001	Phase 1 / Escalation Part / 50000mg
Trial Clinical Indication	CANCER			
Trial Treatment	Drug A		Oral 50000mg 10Q4W	
Relevant Concomitant Medications				
CM1 ORAL 4000 mg, from Day -200 to Day 2; for AE1				
CM2 ORAL 20mg QID, from Day -10 to Day 21; for AE2				
CM3 ORAL 40 mg, from Day -6 to Ongoing; for AE3				
CM4 ORAL 2g BID, from Day 2 to Ongoing; for AE4				
CM5 ORAL 300mg TID, from Day 2 to Ongoing; for AE5				
CM6 ORAL 500mg BID, from Day 5 to Day 10; for AE6				

Narrative Output

Display 3. Pictorial Overview of Narrative Generation

but can be easily done using our new method. As described above, tag the cells and rows with unique codes. Make sure no codes are the same for different cells and rows. For the value of concomitant medications, we need to put all qualified concomitant medications together in one record and list them in the output. For each next concomitant medication add `</w:t><w:br/><w:t>` (`/w:t` is text end element, `w:br` is a line break element and `w:t` is text element in XML) and it will separate each concomitant medication into a new line. The minimized DOCUMENT.XML file is to show how code, #CM001, is updated by the clinical data in the SAS data set.

EXAMPLE 2

As described above our new method uses unique codes to tag clinical data in SAS data set and where they should be put in the narrative template and then exchanged them in DOCUMENT.XML. For automation of a large scale of narratives it is more efficient to have just one template to work for all subjects. It will be straightforward if the number of cells and rows for an event in the template are the same for all subjects. However, some of sections in the template can vary with different subjects. For the example shown in Display 4. Different subjects may have different number of serious adverse events (SAEs) and require different number of rows for this section. Display 4 shows the process of how we add additional rows using programming. Firstly, create just one row in template and tag the cells in the row with continuous numbers, for example, tag the cell for adverse event with '#PT001'. Secondly, create SAS data set using the standard process but name the codes using continuous numbers, for example if there are three SAEs then name them, '#PT001', '#PT002' and '#PT003'. Thirdly, add rows in DOCUMENT.XML by programming. From SAS data set we know how many SAEs for a subject, that is also how many rows are needed in DOCUMENT.XML in order to present all SAEs. As shown in Display 4, the coded template has total three rows, the first two are for table headers, the third row is coded for clinical data. DOCUMENT.XML shows the same structure, there are three row elements, `w:tr` and 7 cell elements `w:tc` under the third row element. Most of elements in DOCUMENT.XML are minimized and only the last cell is expanded to show in Display 4. To add additional rows, we need to replicate all elements of the third row in DOCUMENT.XML and the times of replication is determined by the number of SAEs in SAS data set. For each addition row codes of each cell will be updated by +1, for example, update '#PT001' to '#PT002'. In the example shown in Display 4 we add two rows and update codes for cells by +1 for each addition. As you can see in updated DOCUMENT.XML there are now 5 row elements, `w:tr` and 7 cell elements `w:tc` for the three coded rows. Most of element in updated DOCUMENT.XML are minimized and only the last cell is expanded to show in Display 4. Fourthly, read the updated DOCUMENT.XML along with SAS data set into SAS and follow the same standard procedure discussed above to generate narrative output. This procedure requires an extra step to replicate row elements which may be difficult for people who are not familiar with XML structure. The

of part of narrative, we have used one template and one data set to process more than one hundred of narratives.

REFERENCES

[1] ICH E3 Structure and content of clinical study reports. Published on 01/07/1996.

[2] Hoyle Larry. "Reading Microsoft Word XML files with SAS", SUGI 31, 2006.

[3] Girase Prafulla, Boisvert Daniel, Kurnala Vineeth. "Programmed Patient Narrative Using SAS", PhUSE 2016

[4] zip file from a specific type in a folder. <https://communities.sas.com/t5/SAS-Programming/zip-files-from-a-specific-type-in-a-folder/td-p/438694>

CONTACT INFORMATION

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

Yuping Wu
PRA Health Science
9755 Ridge Drive
Lenexa, KS 66219, USA
wuyuping@prahs.com

Jan Skowronski
Genmab A/S
Kalvebod Brygge 43
1560 Copenhagen V, Denmark
jsk@genmab.com

Any brand and product names are trademarks of their respective companies.