PharmaSUG China 2025 - Paper AA - 179

Tools for Automating SDTM and ADaM Workflows

Fang Wang, Zhiyuan Hu, Xiaoyu Ding, Wei Chu and Zhuo Huang, Keymed Biosciences Inc.

ABSTRACT

Based on CDISC standards, statistical programmers in clinical projects are primarily responsible for generating deliverables such as Annotated Case Report Form (aCRF), Study Data Tabulation Model(SDTM), Analysis Data Model (ADaM), Define.xml, Data Review Guide (-DRG), and Tables, Figures, and Listings (TFLs) in accordance with SDTM Implementation Guide (SDTMIG), ADaM Implementation Guide (ADaMIG), Statistical Analysis Plan (SAP), and Shell specifications. As projects become larger and more complex, traditional workflows not only result in high labor costs but also become bottlenecks for project efficiency and quality. In this paper, we propose a set of automated workflows covering the whole process of SDTM and ADaM. By integrating SAS, Python, and AI technologies, it achieves automation from aCRF generation, SDTM/ADaM mapping, and Define.xml construction to DRG output, significantly reducing repetitive workloads. Empirical evidence demonstrates that this workflow not only enhances the standardization of data processing but also saves programmers over 50% of their working time.

INTRODUCTION

This workflow leverages the construction of the aCRF Mapping Sheet as a central component, seamlessly integrating aCRF automation, SDTM automation, and ADaM automation across all key stages of the clinical data analysis process. This approach establishes a comprehensive and systematic framework for clinical data programming, as illustrated in Figure 1.

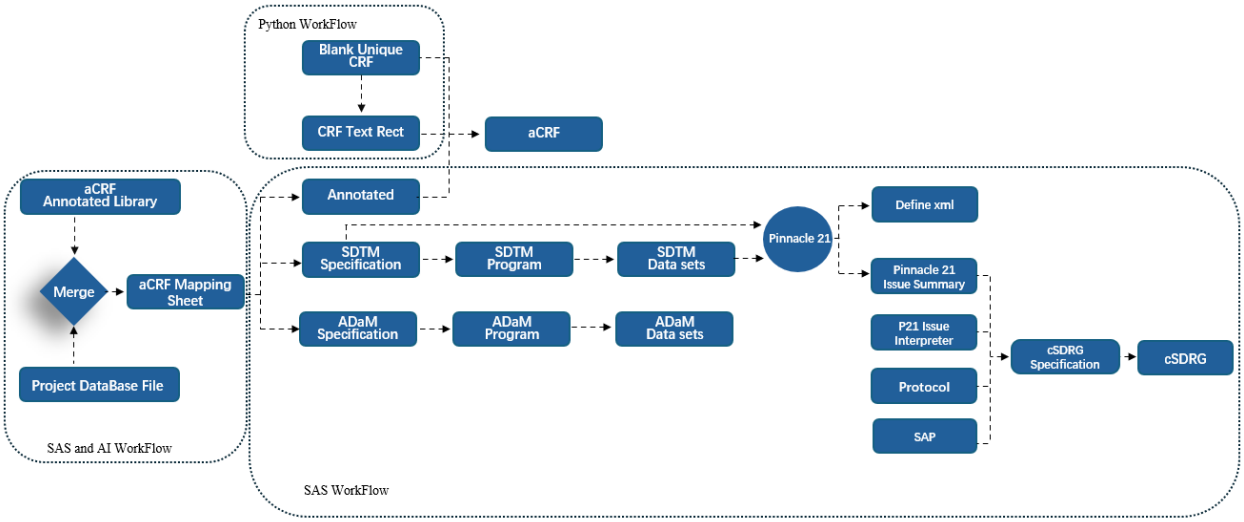

Figure 1 SDTM and ADaM Automated Workflow

This paper is organized into the following seven sections to provide a detailed introduction to the workflow and its practical applications

1. aCRF Mapping Sheet Preparation – The cornerstone of workflow automation.
2. aCRF Automation – Automated generation of aCRF.
3. SDTM Automation – Achieving automated transformation from raw data to the standard model.
4. ADaM Automation – Automated generation of analysis datasets.
5. Define.xml Automation – Automated creation of define.xml.
6. -DRG Automation – Automated construction of the Data Review Guide.

7. Practical Implementation and Application.

1. ACRF MAPPING SHEET PREPARATION

The aCRF mapping sheet is central to the implementation of the entire automation workflow and plays an indispensable role in this process. Its five main functions are as follows:

1. Provide detailed annotation content for aCRF.
2. Supply accurate variable sources for the SDTM Spec.
3. Offer comprehensive variable mapping and splitting references for ADaM automation.
4. Deliver code list and CRF page information for Define.xml generation.
5. Serve as a complete configuration file for automated Listings (TFLs automation).

Figure 2 shows Keymed's aCRF mapping sheet, which mainly includes the following variables:

- FormLabel: Name of the CRF page.
- Form: OID of the CRF form.
- Variable: Annotated CRF variable name.
- Label: CRF field description.
- ItemName: Variable code list.
- Domain: Domain to which the variable belongs.
- StandardName: SDTM standard variable name.
- StandardLabel: Standard label (for SUPP variables only).
- aCRFMappingDescription: aCRF annotation content.
- SDTMType: Indicate whether it is a SUPP variable or a standard variable.

Form	FormLabel	Variable	Label	ItemName	Domain	StandardName	StandardLabel	aCRFMappingDescription	SDTMType
PE	体格检查				PE	DOMAIN		PE(体格检查)	
PE	体格检查	PEPERF	是否进行体格检查？		PE	PESTAT		PESTAT	
PE	体格检查	PEREASND	如选"否"，请说明原因		PE	PEREASND		PEREASND	
PE	体格检查	PEDAT	检查日期		PE	PEDTC		PEDTC	
PE	体格检查	PEPERF1	本次体格检查是否有异常发现？（如是，请详细填写异常		PE	PEORRES		PEORRES=正常 when PETESTCD=INTP	
PE	体格检查	PETEST	异常部位	头面部	PE	PETEST		PETEST when PETESTCD=HEAD	
PE	体格检查	PETEST	异常部位	眼部	PE	PETEST		PETEST when PETESTCD=EYE	
PE	体格检查	PETEST	异常部位	耳鼻喉部	PE	PETEST		PETEST when PETESTCD=THROAT	
PE	体格检查	PETEST	异常部位	口腔	PE	PETEST		PETEST when PETESTCD=ORAL	
PE	体格检查	PETEST	异常部位	皮肤	PE	PETEST		PETEST when PETESTCD=SKIN	
PE	体格检查	PETEST	异常部位	淋巴结	PE	PETEST		PETEST when PETESTCD=LYMPH	
PE	体格检查	PETEST	异常部位	呼吸系统	PE	PETEST		PETEST when PETESTCD=RESP	
PE	体格检查	PETEST	异常部位	心血管系统	PE	PETEST		PETEST when PETESTCD=CARDIO	
PE	体格检查	PETEST	异常部位	腹部	PE	PETEST		PETEST when PETESTCD=ABDOMEN	
PE	体格检查	PETEST	异常部位	泌尿生殖系统	PE	PETEST		PETEST when PETESTCD=URINARY	
PE	体格检查	PETEST	异常部位	肌肉骨骼	PE	PETEST		PETEST when PETESTCD=MUSCLE	
PE	体格检查	PETEST	异常部位	神经系统	PE	PETEST		PETEST when PETESTCD=NERVOUS	
PE	体格检查	PETEST	异常部位	精神状态	PE	PETEST		PETEST when PETESTCD=MENTAL	
PE	体格检查	PETEST	异常部位	其他部位	PE	PETEST		PETEST when PETESTCD=OTHERLOC	
PE	体格检查	PERES	异常评价		PE	PEORRES		PEORRES	
PE	体格检查	PEDESC	异常描述		PE	ABNDESP	异常情况描述	ABNDESP in SUPPPE	SUPP

Figure 2 Example of aCRF Mapping Sheet

1.1 BUILD AN ACRF ANNOTATIONS LIBRARY

Creating aCRF mapping sheets manually for every project would create a lot of repetitive work and result in significant wasted man-hours, so it is essential to create a dictionary library for automatic generation of aCRF Mapping sheets. Figure 3 shows a sample dictionary library.

Form	FormLabel	Variable	Label	ItemName	Domain	StandardName	StandardLabel	aCRFMappingDescription	SDTMType
PE	体格检查				PE	DOMAIN		PE(体格检查)	
PE	体格检查	PEPERF	是否进行体格检查?		PE	PESTAT		PESTAT	
PE	体格检查	PEREASND	如选"否"，请说明原因		PE	PEREASND		PEREASND	
PE	体格检查	PEDAT	检查日期		PE	PEDTC		PEDTC	
PE	体格检查	PEPERF1	本次体格检查是否有异常发现?（如是，请详细填写异常情况）		PE	PEORRES		PEORRES=正常 when PETESTCD=INTP	
PE	体格检查	PETEST	异常部位	一般情况	PE	PETEST		PETEST when PETESTCD=GENERAL	
PE	体格检查	PETEST	异常部位	头面部	PE	PETEST		PETEST when PETESTCD=HEAD	
PE	体格检查	PETEST	异常部位	眼部	PE	PETEST		PETEST when PETESTCD=EYE	
PE	体格检查	PETEST	异常部位	耳鼻喉部	PE	PETEST		PETEST when PETESTCD=THROAT	
PE	体格检查	PETEST	异常部位	口腔	PE	PETEST		PETEST when PETESTCD=ORAL	
PE	体格检查	PETEST	异常部位	颈部	PE	PETEST		PETEST when PETESTCD=NECK	
PE	体格检查	PETEST	异常部位	胸部	PE	PETEST		PETEST when PETESTCD=CHEST	
PE	体格检查	PETEST	异常部位	脊柱	PE	PETEST		PETEST when PETESTCD=SPINE	
PE	体格检查	PETEST	异常部位	四肢	PE	PETEST		PETEST when PETESTCD=LIMBS	
PE	体格检查	PETEST	异常部位	皮肤	PE	PETEST		PETEST when PETESTCD=SKIN	
PE	体格检查	PETEST	异常部位	淋巴结	PE	PETEST		PETEST when PETESTCD=LYMPH	
PE	体格检查	PETEST	异常部位	呼吸系统	PE	PETEST		PETEST when PETESTCD=RESP	
PE	体格检查	PETEST	异常部位	心血管系统	PE	PETEST		PETEST when PETESTCD=CARDIO	
PE	体格检查	PETEST	异常部位	腹部	PE	PETEST		PETEST when PETESTCD=ABDOMEN	
PE	体格检查	PETEST	异常部位	泌尿生殖系统	PE	PETEST		PETEST when PETESTCD=URINARY	
PE	体格检查	PETEST	异常部位	肌肉骨骼	PE	PETEST		PETEST when PETESTCD=MUSCLE	
PE	体格检查	PETEST	异常部位	神经系统	PE	PETEST		PETEST when PETESTCD=NERVOUS	
PE	体格检查	PETEST	异常部位	精神状态	PE	PETEST		PETEST when PETESTCD=MENTAL	
PE	体格检查	PETEST	异常部位	其他部位	PE	PETEST		PETEST when PETESTCD=OTHERLOC	
PE	体格检查	PERES	异常评价		PE	PEORRES		PEORRES	
PE	体格检查	PEDESC	异常描述		PE	ABNDESP	异常情况描述	ABNDESP in SUPPPE	SUPP

Figure 3 Example of aCRF Annotations Library

The construction of the dictionary library directly is related to the matching rate and standardization of the aCRF Mapping Sheet; therefore, so it needs to meet at least the following conditions:

- **Basic Contents of the Dictionary Library:**
    1. As many CRF Form and Variable Names and Labels as possible (used as IDs to match with the project database file).
    2. Standard variables required by the aCRF Mapping Sheet, such as Domain, StandardName, StandardLabel, etc.
- **Basic Functions of the Dictionary Library:**
    1. **Standardization and Compliance:** To ensure standardization from the source, all variable mappings in the dictionary library should comply with SDTMIG. For ADaM and TFLs, mappings should strictly follow the company's conventions.
    2. **Automatic Expansion:** In the early stages of dictionary library building, new entries are inevitable. Setting the dictionary library to automatic expansion mode can greatly enhance the speed of extension. The following flowchart illustrates the concept of automatic expansion for the dictionary library.

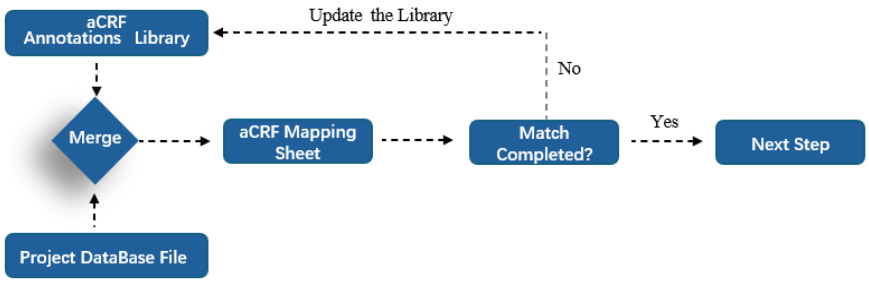

Figure 4 aCRF Annotations Library Update Workflow

1.2 GENERATION OF THE ACRF MAPPING SHEET

Taking the Chinese project in the Taimei EDC system (Figure 5) as an example, by performing an exact match between the FormLabel, Form, Variable, and Label columns in the dictionary library and the

FormOID, FormName, SasFieldName, and ItemName columns in the project database file, all annotations for the project can be retrieved from the aCRF dictionary library. Figure 6 shows the result of an exact match in the ideal case.

FormOID	FormName	SASDatasetName	SASFieldName	ItemName	CodeListOID
PE	体格检查	PE	PEREASND	如选“否”，请说明原因	
PE	体格检查	PE	PEDAT	检查日期	
PE	体格检查	PE	PEPERF1	本次体格检查是否有异常发现？（如是，请详细填写异常情况）	NY=[1|是,2|否,]
PE	体格检查	PE	PETEST	异常部位	PETEST=[1|头面部,2|眼部,3|耳鼻喉部,4|口腔,5|皮肤,6|淋巴结,7|呼吸系统,8|心血管系统,9|腹部,10|泌尿生殖系统,11|肌肉骨骼,12|神经系统,13|精神状态,99|其他部位,]
PE	体格检查	PE	PERES	异常评价	PERES=[1|异常无临床意义,2|异常有临床意义,]
PE	体格检查	PE	PEDESC	异常描述	

Figure 5 Example of Chinese aCRF Database File

Form	FormLabel	Variable	Label	ItemName	Domain	StandardName	StandardLabel	aCRFMappingDescription	SDTMType
PE	体格检查				PE	DOMAIN		PE(体格检查)	
PE	体格检查	PEPERF	是否进行体格检查？		PE	PESTAT		PESTAT	
PE	体格检查	PEREASND	如选"否"，请说明原因		PE	PEREASND		PEREASND	
PE	体格检查	PEDAT	检查日期		PE	PEDTC		PEDTC	
PE	体格检查	PEPERF1	本次体格检查是否有异常发现？（如是，请详细填写异常		PE	PEORRES		PEORRES=正常 when PETESTCD=INTP	
PE	体格检查	PETEST	异常部位	头面部	PE	PETEST		PETEST when PETESTCD=HEAD	
PE	体格检查	PETEST	异常部位	眼部	PE	PETEST		PETEST when PETESTCD=EYE	
PE	体格检查	PETEST	异常部位	耳鼻喉部	PE	PETEST		PETEST when PETESTCD=THROAT	
PE	体格检查	PETEST	异常部位	口腔	PE	PETEST		PETEST when PETESTCD=ORAL	
PE	体格检查	PETEST	异常部位	皮肤	PE	PETEST		PETEST when PETESTCD=SKIN	
PE	体格检查	PETEST	异常部位	淋巴结	PE	PETEST		PETEST when PETESTCD=LYMPH	
PE	体格检查	PETEST	异常部位	呼吸系统	PE	PETEST		PETEST when PETESTCD=RESP	
PE	体格检查	PETEST	异常部位	心血管系统	PE	PETEST		PETEST when PETESTCD=CARDIO	
PE	体格检查	PETEST	异常部位	腹部	PE	PETEST		PETEST when PETESTCD=ABDOMEN	
PE	体格检查	PETEST	异常部位	泌尿生殖系统	PE	PETEST		PETEST when PETESTCD=URINARY	
PE	体格检查	PETEST	异常部位	肌肉骨骼	PE	PETEST		PETEST when PETESTCD=MUSCLE	
PE	体格检查	PETEST	异常部位	神经系统	PE	PETEST		PETEST when PETESTCD=NERVOUS	
PE	体格检查	PETEST	异常部位	精神状态	PE	PETEST		PETEST when PETESTCD=MENTAL	
PE	体格检查	PETEST	异常部位	其他部位	PE	PETEST		PETEST when PETESTCD=OTHERLOC	
PE	体格检查	PERES	异常评价		PE	PEORRES		PEORRES	
PE	体格检查	PEDESC	异常描述		PE	ABNDESP	异常情况描述	ABNDESP in SUPPPE	SUPP

Figure 6 Matching Results Between the Database File and the Annotations Library

In practice, due to differences in field descriptions, the matching rate between the database and the dictionary library may be quite low, generally less than 50%. If both FormOID and Variable are matched simultaneously, the matching rate will be further reduced. For information that cannot be matched, although complete matching can be achieved by manually expanding the dictionary library, this approach will lead to an increasingly large dictionary that still cannot meet the needs of all projects, especially for Chinese and English projects, where it may be necessary to establish two separate aCRF dictionary libraries.

We can improve the matching rate through the following two approaches:

1. Cross-department collaboration to solve the problem of OID inconsistency from the source:

In order to prevent FormOID and Variable OID from decreasing the matching rate, we have been cooperating deeply with the DM department since the beginning of the establishment of the database, and intervened in the database construction process of the DM, restricting the use of the OID with the dictionary, so that all the OIDs of the project will try to use the contents that have already been included in the dictionary. This operation can increase the matching rate by at least 30%.

2. Introducing AI to solve the problem of inconsistent field descriptions from a semantic perspective:

Introducing AI to solve the problem of inconsistent field descriptions from a semantic perspective: for those who have already restricted OID, but cannot match because of inconsistent field descriptions, the introduction of AI completely solves the difficulties for us. The semantic matching function of AI not only matches fields in the same language but also performs bilingual matching. The following is example code for semantic matching using AI and Figure 7 demonstrates the effect of matching an English project using a Chinese dictionary library, Figure 6 demonstrates an English project database file.

System Prompt:

You are a clinical data standards expert. Your task is to match clinical trial fields to a known dictionary library.

For each item, suggest:

- Domain

- StandardName

- StandardLabel

- aCRFMappingDescription

Only choose from the library list provided. Don't create new variable names. If no good match is found, return:

- Domain: "Unmapped"

- StandardName: " "

- StandardLabel: " "

- aCRFMappingDescription: " "

Core Code:

```
 user_prompt = f"""Here are the {language} clinical fields to map:\n{json.dumps(input_df.to_dict(orient="records"), ensure_ascii=False, indent=2)}

Use the following SDTM Library for matching:\n {json.dumps(library_data, ensure_ascii=False, indent=2)}"""

   response = client.chat.completions.create(
      model="qwen3-235b-a22b",  # or "deepseek-v3"
      messages=[
         {"role": "system", "content": system_prompt},
         {"role": "user", "content": user_prompt}
      ],
      temperature=0.2
   )
```

FormOID	Ordinal	VariableOID	DataFormat	PreText	DefaultValue
PE_01	1	PEPERF	$2	Was the examination performed?	
PE_01	2	PEREASND	$200	If no, please specify the reason	
PE_01	3	PEDAT	yyyy/mm/dd	Examination date	
PE_01	4	PETEST	$20	Body System	GENERAL_CONDITION|SKIN|LYMPH NODES|HEAD|NECK|CHEST|ABDOMEN|SPINE|LIMBS|OTHER|NERVOUS SYSTEM|
PE_01	5	PERES	$6	Result	
PE_01	6	PEDESC	$1999	If abnormal, please specify	
PE_01	7	NOW	dd MMM yyyy HH:nn:ss	NOW	

Figure 6 Example of English aCRF Database File

FormOID	FormLabel	Variable	Label	ItemName	Domain	StandardName	StandardLabel	aCRFMappingDescription
PE_01	Physical Examination	PEPERF	Was the examination performed?		PE	PESTAT		PESTAT
PE_01	Physical Examination	PEREASND	If no, please specify the reason		PE	PEREASND		PEREASND
PE_01	Physical Examination	PEDAT	Examination date		PE	PEDTC		PEDTC
PE_01	Physical Examination	PETEST	Body System	General condition	PE	PETEST		PETEST when PETESTCD=GENERAL
PE_01	Physical Examination	PETEST	Body System	Skin	PE	PETEST		PETEST when PETESTCD=SKIN
PE_01	Physical Examination	PETEST	Body System	Systemic superficial lymph nodes	PE	PETEST		PETEST when PETESTCD=LYMPH
PE_01	Physical Examination	PETEST	Body System	Head	PE	PETEST		PETEST when PETESTCD=HEAD
PE_01	Physical Examination	PETEST	Body System	Neck	PE	PETEST		PETEST when PETESTCD=NECK
PE_01	Physical Examination	PETEST	Body System	Chest	PE	PETEST		PETEST when PETESTCD=CHEST
PE_01	Physical Examination	PETEST	Body System	Abdomen	PE	PETEST		PETEST when PETESTCD=ABDOMEN
PE_01	Physical Examination	PETEST	Body System	Spine	PE	PETEST		PETEST when PETESTCD=SPINE
PE_01	Physical Examination	PETEST	Body System	Limbs	PE	PETEST		PETEST when PETESTCD=LIMBS
PE_01	Physical Examination	PETEST	Body System	Other	PE	PETEST		PETEST when PETESTCD=OTHERLOC
PE_01	Physical Examination	PETEST	Body System	Nervous System	PE	PETEST		PETEST when PETESTCD=NERVOUS
PE_01	Physical Examination	PERES	Result		PE	PEORRES		PEORRES
PE_01	Physical Examination	PEDESC	If abnormal, please specify		PE	ABNDESP	异常情况描述	ABNDESP in SUPPPE
PE_01	Physical Examination	NOW	NOW		Unmapped			

Figure 7 Matching Results Between the English Database File and the Chinese Dictionary Library

2. ACRF AUTOMATION

2.1 PREPARATION FOR ACRF AUTOMATION:

1. aCRF Mapping Sheet.
2. Unique Blank CRF.
3. CRF Field Rect: Python offers several packages for this task, such as PDFMiner, PyMuPDF, and PyPDF2.

2.2 IMPLEMENTATION OF ACRF AUTOMATION:

Merge the CRF field information and aCRF Mapping sheet to generate an .xfdf file and import it to Unique Blank CRF file, then you can complete the aCRF production. The following is an example of the generated aCRF in Figure 8.

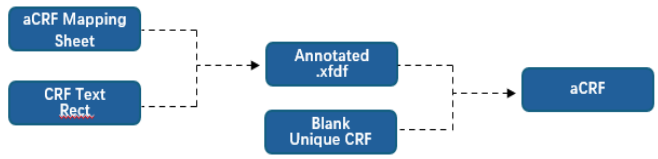

Figure 8 aCRF Annotations Workflow

The following are examples of the generated aCRF:

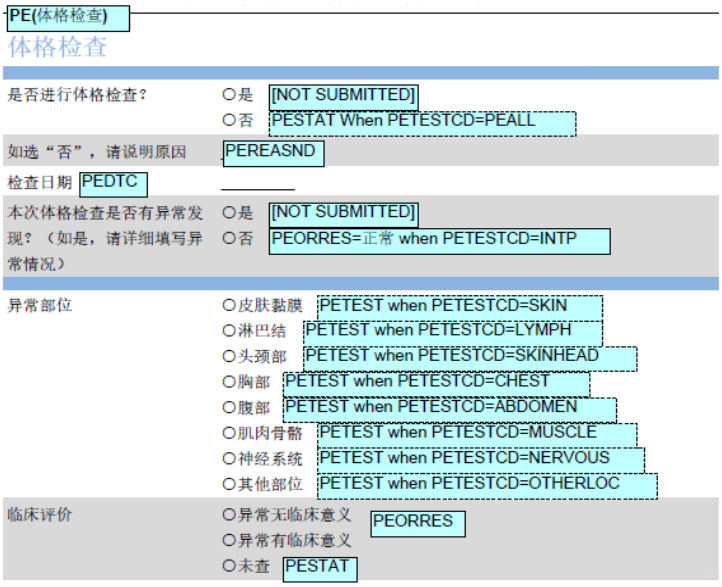

Figure 9: aCRF generated from a Chinese project Database File

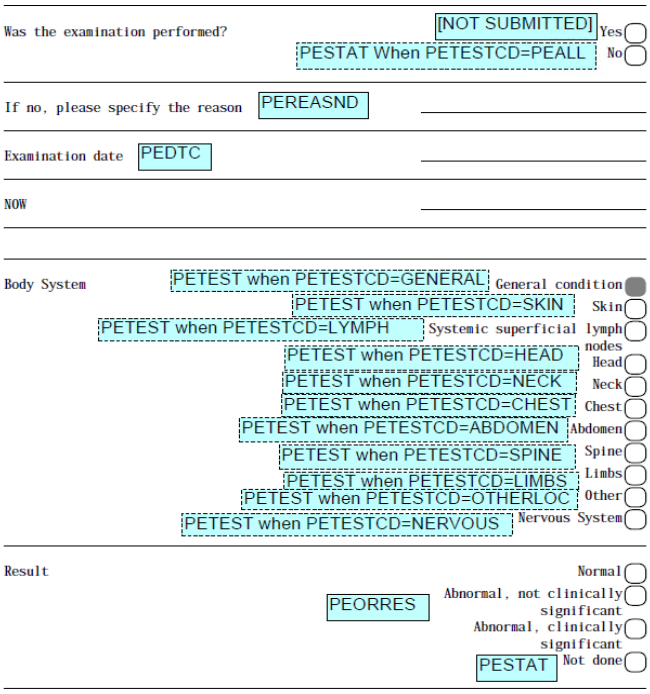

Figure 10: aCRF Generated from an English Project Database file

2.3 PROBLEMS AND SOLUTIONS IN THE PROCESS OF ACRF AUTOMATION

Although the above examples demonstrate solutions to aCRF matching problems and ideal output results, there are still many details that need to be paid attention to in practice, such as:

1. Categorical Mapping of Variable Values

- Problem: For example, in physical examination clinical evaluation, Value=NOT DONE needs to be mapped to PESTAT, and the rest of the values need to be mapped to PEORRES or other variables, and the dictionary library does not contain such detailed mappings.

- Solution: When you need to map the variable values separately, you can configure the logic to traverse the list of values in the program. For the values that need to be mapped separately, they should be automatically categorized into specific groups, such as -STAT and DSDECOD.

2. Matching Failure Due to Line Breaks When Extracting Rect

- Problem: When grabbing the location information in Python, if the field happens to be line breaks, it will be considered as two fields, which will lead to failure to match the database file. If approximate matching is used, it will lead to duplicate results.
- Solution: Use the database file as a dictionary library to restore the truncated field to complete field before performing an exact match.

3. Duplicate Annotations Caused by Repeated field

- Problem: There may be cases in aCRF where the same field appears multiple times but maps to different objects. For example, a concomitant medication page may contain multiple "Other, please specify" fields, but each refers to a different object. If left unprocessed, each field will generate multiple annotation field boxes.
- Solution: Determine the occurrence order of duplicate fields, assign a unique ID to each, and then perform an exact match using the field IDs.

3. SDTM AUTOMATION

3.1 APPROACH TO SDTM AUTOMATION

SDTM datasets have relatively stable structures, and their main task is to map raw data. The mapping relationship has been realized in the first step of the aCRF automation process, so in this step, we only need to automatically output SDTM Spec according to the aCRF mapping sheet generated in the first step to complete Spec automation. After the Spec automation is completed, the Spec can be interpreted and split by the program, and the results can be further used to automatically generate the program (manual review and revision are still required after automation, so it is recommended to automate the program based on the Spec rather than directly on the aCRF mapping sheet).

3.2 SDTM SPECIFICATION GENERATION

Preparation for SDTM Spec automation:

1. aCRF mapping sheet.
2. SDTM Spec template: The template structure should comply with SDTMIG requirements and include derivation methods for derived variables (using general methods to describe the logic can avoid repeated modifications across different projects).

Implementation of Spec automation:

The aCRF mapping sheet contains all the standard variables from CRF and their sources, --TESTCD, --TEST mapping relationship with the original data and code list value, according to specific rules to integrate this information and merge them with the SDTM Spec template to get the complete SDTM Spec. As shown in Figure 12 and Figure13.

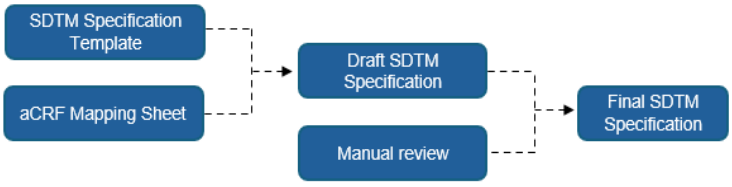

Figure 11 Approach to SDTM Specification Automation

Name	Label_en	Typ	Cor	Length	Controlled Terms, Codelist or Form	Origin	Source	Derivation/Comment	Includ
STUDYID	Study Identifier	Char	Req	200		Protocol		DM.STUDYID	Y
DOMAIN	Domain Abbreviation	Char	Req	200	DOMAIN	Assigned		Set to "AE"	Y
USUBJID	Unique Subject Identifier	Char	Exp	200		Derived		DM.USUBJID	Y
AESEQ	Sequence Number	Num	Req	8		Derived		After sorting by the Key variable, the first record for the same subject is assigned a value of 1, and subsequent records are incremented by 1	Y
AESPID	Sponsor-Defined Identifier	Char	Perm	200		eDT	RAW.AE.RECREP,VISTREP; RAW.ARIA1.RECREP,VISTREP; RAW.IRR.RECREP,VISTREP		Y
AETERM	Reported Term for the Adverse Event	Char	Req	200		CRF	RAW.AE.AETERM; RAW.ARIA1.AR1TERM; RAW.IRR.IRRTERM		Y
AELLT	Lowest Level Term	Char	Exp	200	MedDRA	eDT	RAW.AE.LLTTERM; RAW.ARIA1.LLTTERM; RAW.IRR.LLTTERM		Y
AELLTCD	Lowest Level Term Code	Num	Exp	8	MedDRA	eDT	RAW.AE.LLTCODE; RAW.ARIA1.LLTCODE; RAW.IRR.LLTCODE		Y
AEDECOD	Dictionary-Derived Term	Char	Req	200	MedDRA	eDT	RAW.AE.PTTERM; RAW.ARIA1.PTTERM; RAW.IRR.PTTERM		Y
AEPTCD	Preferred Term Code	Num	Exp	8	MedDRA	eDT	RAW.AE.PTCODE; RAW.ARIA1.PTCODE; RAW.IRR.PTCODE		Y
AEHLT	High Level Term	Char	Exp	200	MedDRA	eDT	RAW.AE.HLTTERM; RAW.ARIA1.HLTTERM; RAW.IRR.HLTTERM		Y
AEHLTCD	High Level Term Code	Num	Exp	8	MedDRA	eDT	RAW.AE.HLTCODE; RAW.ARIA1.HLTCODE; RAW.IRR.HLTCODE		Y
AEHLGT	High Level Group Term	Char	Exp	200	MedDRA	eDT	RAW.AE.HLGTTERM; RAW.ARIA1.HLGTTERM; RAW.IRR.HLGTTERM		Y
AEHLGTCD	High Level Group Term Code	Num	Exp	8	MedDRA	eDT	RAW.AE.HLGTCODE; RAW.ARIA1.HLGTCODE; RAW.IRR.HLGTCODE		Y
AECAT	Category for Adverse Event	Char	Perm	200	AECAT	Assigned		Assign values according to the following rules based on the data source: RAW.AE-->"Adverse Events"; RAW.ARIA1-->"ARIA"; RAW.IRR-->"IRR"	Y

Figure 12 Example of SDTM.AE Specification

Name	Label_en	Type	Core	Length	Controlled Terms, Codelist or Format	Origin	Source	Derivation/Comment	Include
STUDYID	Study Identifier	Char	Req	200		Protocol		DM.STUDYID	Y
DOMAIN	Domain Abbreviation	Char	Req	200	DOMAIN	Assigned		Set to "EG"	Y
USUBJID	Unique Subject Identifier	Char	Exp	200		Derived		DM.USUBJID	Y
EGSEQ	Sequence Number	Num	Req	8		Derived		After sorting by the Key variable, the first record for the same subject is assigned a value of 1, and subsequent records are incremented by 1	Y
EGSPID	Sponsor-Defined Identifier	Char	Perm	200		eDT	RAW.EG.RECREP,VISTREP		Y
EGTESTCD	ECG Test or Examination Short Name	Char	Req	200	EGTESTCD	Assigned		Reference to MetaValue Sheet	Y
EGTEST	ECG Test or Examination Name	Char	Req	200	EGTEST	Assigned		Reference to MetaValue Sheet	Y
EGORRES	Result or Finding in Original Units	Char	Exp	200		CRF	RAW.EG.EGHR; RAW.EG.EGPR; RAW.EG.EGQRS; RAW.EG.EGQT; RAW.EG.EGQTCF; RAW.EG.EGCLSIG		Y
EGORRESU	Original Units	Char	Perm	200	EG_UNIT	CRF		Assign values according to the following rules based on the data source: RAW.EG.EGHR-->"bpm"; RAW.EG.EGPR-->"ms"; RAW.EG.EGQRS-->"ms"; RAW.EG.EGQT-->"ms"; RAW.EG.EGQTCF-->"ms"	Y
EGSTRESC	Character Result	Char	Exp	200		Assigned		EGORRES in standardized units	Y
EGSTRESN	Numeric Result	Num	Perm	8		Assigned		numeric value of EGSTRESC.	Y
EGSTRESU	Standard Units	Char	Perm	200	EG_UNIT	Assigned		Reference to MetaUnit Sheet	Y
EGSTAT	Completion Status	Char	Perm	200	ND	CRF		Assign values according to the following rules based on the data source: RAW.EG.EGPERF="No"-->"NOT DONE"	Y
EGREASND	Reason ECG Not Performed	Char	Perm	200		CRF	RAW.EG.EGREASND		Y
EGBLFL	Baseline Flag	Char	Exp	200	YNULL	Derived		Assigned as "Yes" to the last non-empty record prior to the first administration of the study drug.	Y
VISITNUM	Visit Number	Num	Exp	8	VISITNUM	Assigned		Reference to SDTM.SV	Y
VISIT	Visit Name	Char	Perm	200	VISIT	Assigned	RAW.EG.VISIT	Reference to SDTM.SV	Y
EGDTC	Date	Char	Exp	200	ISO 8601	CRF	RAW.EG.EGDAT		Y

Figure 13 Example of SDTM.EG Specification

3.3 CONVERT SDTM SPECIFICATION TO SDTM PROGRAM

The core of SDTM program automation is to enable SAS programs to accurately parse the mapping rules defined in the SDTM Spec. The rules are configured with a set of compilation logic that allows the tool to automatically compile the program according to the logic. An example of a mapping rule is shown in Figure 14.

Figure 14 Approach to SDTM Program Automation

- **Semicolon (;)**: Indicate that a standard variable has multiple independent sources (requiring concatenation of raw datasets).
- **Comma (,)**: Indicate that a standard variable is formed by combining values from multiple raw variables (requiring value merging).
- **Arrow (→)**: Indicate a dependency or mapping relationship between standard variables.

Taking AE. AETERM (Figure 15) and EG. EGORRES (Figure 16) as examples to illustrate the corresponding parsing logic:

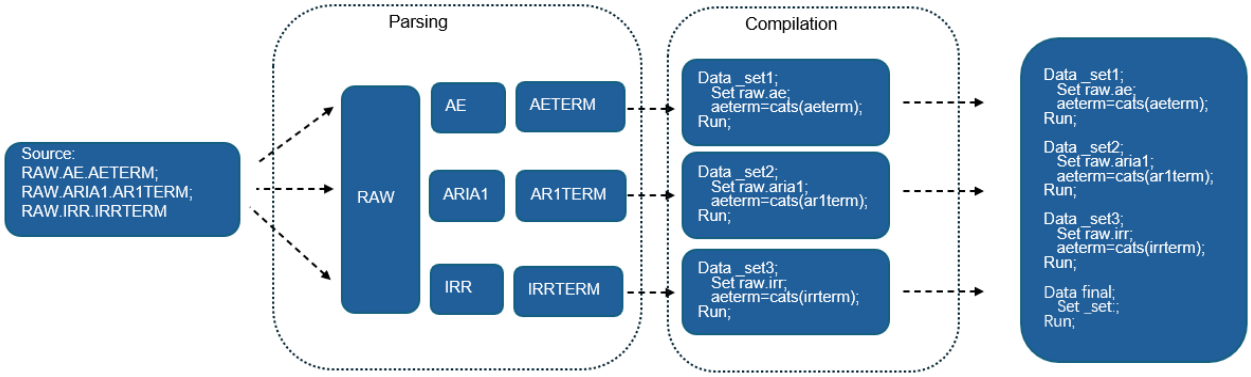

Figure 15 Example of SDTM.AE Program Automation

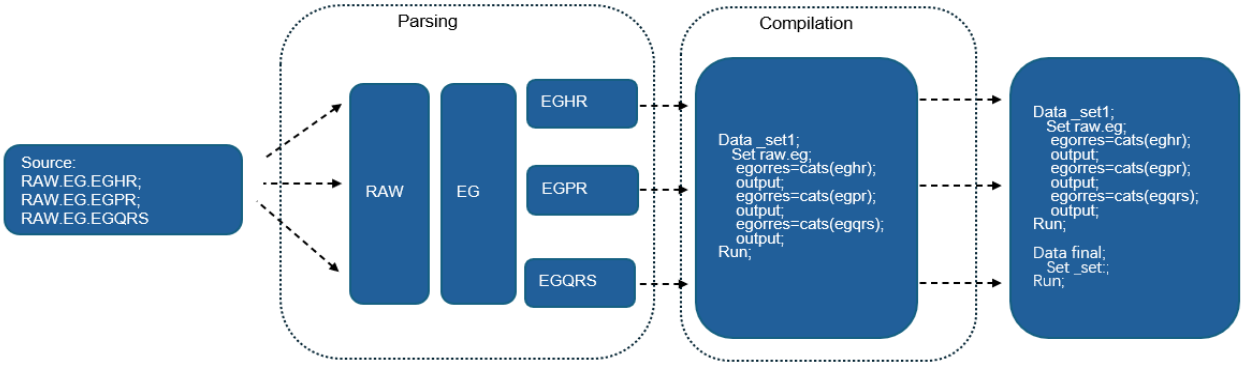

Figure 16 Example of SDTM.EG Program Automation

Based on the precise interpretation of SDTM Spec mapping relationships, the tool can efficiently automate the generation (complete output examples in Figure 16 and Figure 17), and well-configured mapping rules and supporting logic can make the automated output of the program to a higher degree of perfection.

```
data _set1;
     Length subjid aespid aeterm aellt aedecod aehlt aehlgt aecat aebodsys ... $200. ;
     set raw.ae(rename=(aeacn=_aeacn aedis=_aedis aeout=_aeout aerel=_aerel ...));
     subjid=cats(_subjid);
     aespid=catx('-',recrep,vistrep);
     aecat="Adverse Events";
     aestdtc=dtm(aestdat,aesttim);
     aeendtc=dtm(aeendat,aeentim);
     aeterm=cats(  aeterm);
     aellt=cats(lltterm);
     aelltcd=input(lltcode,??best.);
     aedecod=cats(ptterm);
     aeptcd=input(ptcode,??best.);
     aehlt=cats(hltterm);
     aehltcd=input(hltcode,??best.);
     aehlgt=cats(hlgtterm);
     aehlgtcd=input(hlgtcode,??best.);
     aebodsys=cats(socterm);
     aebdsycd=input(soccode,??best.);
     aesoc=cats(socterm);
     aesoccd=input(soccode,??best.);
     aesev=cats(_aesev);
     aeser=cats(  aeser);
     aeacn=cats(  aeacn);
     aerel=cats(_aerel);
     aeout=cats(_aeout);
     aescong=cats(  aescong);
     aesdisab=cats(_aesdisab);
     aesdth=cats(_aesdth);
     aeshosp=cats(_aeshosp);
     aeslife=cats(  aeslife);
     aesmie=cats(_aesmie);
     ariany=cats(aeirryn);
     irrny=cats(aeariayn);
     contrtyp=cats(aethyn);
     aedis=cats(_aedis);
     dictver=cats(mdraver);
    keep subjid aespid aeterm aellt aelltcd aedecod aeptcd aehlt aehltcd aehlgt aehlgtcd aecat ...;
run;

data   set2;
     Length subjid aespid aeterm aellt aedecod aehlt aehlgt aecat aebodsys aesoc aesev ..... $200.
     set raw.aria1(rename=(subjid=_subjid));
     subjid=cats(_subjid);
     aespid=catx('-',recrep,vistrep);
     aecat="ARIA";
     aestdtc=dtm(ar1stdat,ar1sttim);
     aeendtc=dtm(ar1endat,ar1entim);
     ...;
     keep subjid aespid aeterm aellt aelltcd aedecod aeptcd aehlt aehltcd aehlgt aehlgtcd ...;
run;

data _set3;
     Length subjid aespid aeterm aellt aedecod aehlt aehlgt aecat aebodsys ... $200. ;
     set raw.irr(rename=(subjid=_subjid));
     subjid=cats(  subjid);
     aespid=catx('-',recrep,vistrep);
     aecat="IRR";
     aestdtc=dtm(irrstdat,irrsttim);
     ------;
     keep subjid aespid aeterm aellt aelltcd aedecod aeptcd aehlt aehltcd aehlgt ...;
run;

Data set_final;
     set _set:;
     length studyid usubjid $200.;
     studyid="&  studyid.";
     usubjid=cats("&_studyid.","-",subjid);
     domain="%upcase(&domain.)";
     if aeterm="" then delete;
run;
```

Figure 17 Example of Automated Output AE Program (Part of the code in the Display is replaced by "...")

```
data _set1;
    Length subjid egspid egorres rawtest egorresu egstat egreasnd visit egdtc $200. aeno mhno abndesp $2000.;
    set raw.eg(rename=(egreasnd=_egreasnd subjid=_subjid visit=_visit));
    subjid=cats(_subjid);
    egspid=catx('-',recrep,vistrep);
    visit=cats(_visit);
    egdtc=dtm(egdat,"");
    if egperf="No" then do;
        egstat="NOT DONE";
        egreasnd=cats(_egreasnd);
        rawtest="ALL ECG";
    end;
    else do;
        egorres=cats(eghr);
        rawtest="Heart rate";
        egorresu="bpm";
        output;
        egorres=cats(egpr);
        rawtest="PR interval";
        egorresu="ms";
        output;
        egorres=cats(egqrs);
        rawtest="QRS duration";
        egorresu="ms";
        output;
        egorres=cats(egqt);
        rawtest="QT interval";
        egorresu="ms";
        output;
        egorres=cats(egqtcf);
        rawtest="QTcF";
        egorresu="ms";
        output;
        egorres=cats(egclsig);
        egorresu="";
        rawtest="Clinical significance";
        aeno=cats(egaeno);
        mhno=cats(egmhno);
        abndesp=catx(',',egdesc,egcom);
        output;
    end;
    keep subjid egspid egorres rawtest egorresu egstat egreasnd visit egdtc aeno mhno abndesp;
run;

Data set_final;
    set _set:;
    length studyid usubjid $200.;
    studyid="&_studyid.";
    usubjid=cats("&_studyid.","-",subjid);
    domain="%upcase(&domain.)";
    if rawtest="" then delete;
run;
```

Figure 18 Example of Automated Output EG Program

3.4 AUTOMATE THE DERIVED VARIABLES

In the previous step, we have completed the coding for all variables except derived variables. Based on the stable structure of SDTM, most derived variables are generated using fixed logic, such as --DY, EPOCH, --BLFL, VISIT, VISITNUM, etc. For such variables, it is recommended to use standardized macros in each project and apply them to each domain. This approach can significantly improve the automation efficiency of derived variables, so further details are not discussed here.

3.5 APPLICATION IN KEYMED

According to Keymed's application, Table 1 shows the automation degree of SDTM by Class across five dimensions.

	Trial Design	Special Purpose	Events	Interventions	Findings
Code Completeness	NA	90% (Content involving external data requires manual review or modification)	100%	100%	100%
Manual Intervention Required	100%	10%	0%	0%	20% (Content involving external data requires manual review or modification)
Accuracy of Output	NA	100%	100%	100%	100%
Readability and Compliance	NA	Good	Good	Good	Good
Maintenance Cost	NA	Little	None or minimal	None or minimal	Little

Table 1 Summarize the level of SDTM automation by Class

4. ADAM AUTOMATION

4.1 APPROACH TO ADAM AUTOMATION

Due to the complexity and variability of ADaM, ADaM automation requires more reliance on derivation logic, functional macros, and manual verification, resulting in a level of automation that is not as complete as SDTM.

automatically generate ADaM Specification

As shown in Figure 19, the variables required for the ADaM Spec need to be automatically derived based on the SDTM Spec, SDTM datasets, and the aCRF mapping Sheet. For variables that cannot be automatically derived, manual supplementation is still necessary. Therefore, the better the derivation logic is in this process, the better the automation results are.

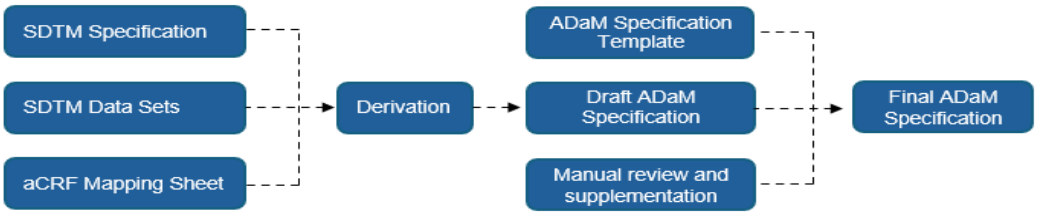

Figure 19 Approach to ADaM Specification Automation

Below are several examples of automatically derived variables:

1. Deriving ADTM, ADT, and ATM variables in ADaM from --DTC variables in SDTM.
2. Mapping --TEST and --TESTCD in SDTM to PARAM, PARAMN, and PARAMCD in ADaM.
3. Mapping DSDECOD in SDTM to DCTREAS and DCSREAS in ADSL.
4. Deriving AVAL, BASE, CHG, PCHG in ADaM from --STRESN variables in SDTM.

For variables that cannot be derived, automatic output can be set based on the attributes of the ADaM

dataset, such as:

1. TRTEMFL and PREFL are automatically output in ADAE.
2. ONTRTFL and PREFL are automatically output in ADCM and ADPR.
3. If the SDTM data contains clinical assessment results, DTYPE is output by default in the ADaM Spec, with the default derivation rule set to WOCF.

If the content of the Specification still does not meet analysis requirements, manual additions are needed.

4.2 ADAM PROGRAM AUTOMATION

Same as SDTM automation program implementation logic, ADaM program also needs to split and decode ADaM Spec implementation, but different from SDTM, ADaM program needs to introduce number processing logic to the parsed content in order after parsing Spec.

Figure 20 Approach to ADaM Program Automation

For example:

If the ADaM Spec indicates that a CHG variable is needed, the logic CHG = AVAL - BASE needs to be introduced, and before that, the derivation logic for the AVAL and BASE variables needs to be supplemented.

The introduced logic should be prepared in advance as functional macros, which can be called as needed in each project. However, in the final output ADaM program, these macros should be parsed into concise and clear SAS code.

4.3 APPLICATION IN KEYMED

A Currently, ADaM automation is still under development. The degree of automation is summarized in Table 2 as follows:

	ADSL	OCCDS	BDS
Code Completeness	50% (Non-standard variables require manual code addition)	100%	50% (Routine analysis datasets such as ADEG, ADLB, ADVS, ADPE, ADPC, ADPP, etc., can be fully automated, but efficacy analysis datasets involving complex imputation or processing still require manual generation of SAS code.)
Manual Intervention Required	50%	0%	50%
Accuracy of Output	100% (Only for automatically generated code)	100%	100% (Only for automatically generated code)
Readability and Compliance	The same as the traditional workflow	Good	The same as the traditional workflow
Maintenance Cost	The same as the traditional workflow	Little	The same as the traditional workflow

Table 2 Summarize the level of ADaM automation by Class

5. DEFINE AUTOMATION

5.1 APPROACH TO DEFINE.XML AUTOMATION

As shown in Figure 21, the generation of Define.xml mainly relies on the SDTM/ADaM Spec and dataset generated in the previous steps:

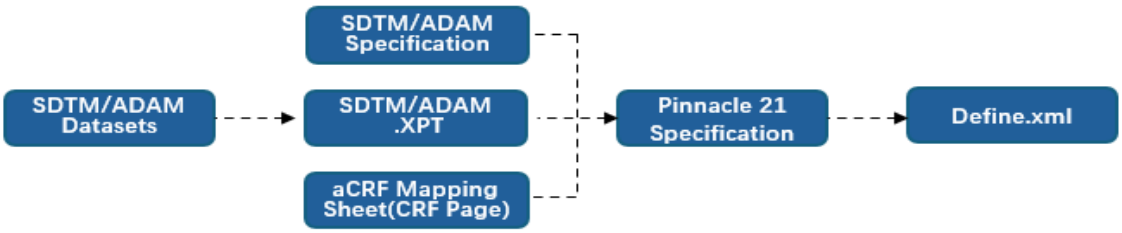

Figure 21 Define.xml Automation Workflow

5.2 PREPARATION FOR DEFINE.XML

1. SDTM/ADaM Spec: Obtain code list and derivation rules.
2. SDTM/ADaM XPT: Obtain actual variables, attributes, and VLM.
3. aCRF Mapping Sheet: Obtain CRF pages information.

5.3 IMPLEMENTATION OF DEFINE.XML

Generate a complete Pinnacle 21 specification by using a tool that integrates SDTM/ADaM spec, SDTM/ADaM xpt and CRF Page information, and import the Pinnacle 21 specification into Pinnacle 21 to automatically generate define.xml. Using tools to integrate information from the SDTM/ADaM spec, SDTM/ADaM xpt, and CRF pages, a complete Pinnacle 21 specification can be generated. Importing the Pinnacle 21 specification into Pinnacle 21 enables the automatic generation of define.xml.

6. -DRG AUTOMATION

6.1 APPROACH TO DRG AUTOMATION

The Data Reviewer's Guide (DRG), as a regulatory submission document, requires extensive manual effort for narrative content creation and formatting adjustments. As shown in Figure 22, the core methodology utilizes SAS programs to systematically extract and organize information required for generating the DRG document. This information is processed iteratively and output to a PDF file. The requisite data originates from three primary sources:

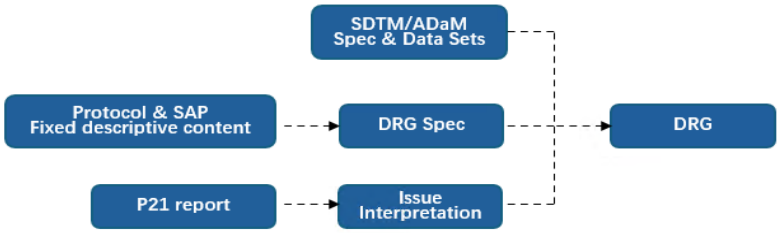

Figure 22 DRG Workflow

1. DRG Specification (Spec): Contains fixed descriptive content (e.g., ADRG Chapters in Figure 23, Section 1.1 Purpose in Figure 24), information derived from the Statistical Analysis Plan (SAP) (e.g., time windows, imputation rules), and details provided by the study protocol (e.g., protocol number, study design).

1. 介绍	1.1 目的
1. 介绍	1.2 缩略语
1. 介绍	1.3 研究数据标准和词典清单
2. 方案描述	2.1 方案编号和标题
2. 方案描述	2.2 与ADaM概念相关的方案设计
3. 与多个分析数据集相关的分析考虑	3.1 SDTM和ADaM的内容比较
3. 与多个分析数据集相关的分析考虑	3.2 核心变量
3. 与多个分析数据集相关的分析考虑	3.3 治疗变量
3. 与多个分析数据集相关的分析考虑	3.4 需要特定分析规则的受试者问题
3. 与多个分析数据集相关的分析考虑	3.5 使用访视时间窗、计划外访视和记录选择
3. 与多个分析数据集相关的分析考虑	3.6 填补/衍生方法
4. 分析数据集的创建和处理问题	4.1 拆分数据集
4. 分析数据集的创建和处理问题	4.2 数据来源
4. 分析数据集的创建和处理问题	4.3 中间数据集
5. 分析数据集描述	5.1 概述
5. 分析数据集描述	5.2 分析数据集
6. 数据符合性汇总	6.1 符合性说明
6. 数据符合性汇总	6.2 问题汇总
7. 程序递交	

Figure 23 ADRG Chapters in Spec

本文档提供了分析数据集和术语的背景知识，是对数据定义文档 (define.xml)的额外补充。 此外，本文档还提供了 ADaM 一致性验证发现的汇总。

Figure 24 Purpose in ADRG Spec

2. SDTM/ADaM Specifications and Datasets: Extracts project-specific content required for the DRG's Data Description section, including domain-level summaries, Supplemental Qualifiers (SUPPQUAL) (encompassing CRF-designed variables not collected in raw data), core variables, and derivation rules for analysis variables.
3. Pinnacle 21 (P21) Reports & Interpretations: Integrates findings from P21 validation reports with standardized explanations (construction of the P21 Issue Interpretation Library detailed in Section 5.2).

6.2 BUILD A PINNACLE 21 ISSUE INTERPRETATION LIBRARY

To enhance the efficiency and consistency of processing the P21 validation results of datasets generated by SDTM/ADaM automated tools, we have compiled a centralized P21 issue interpretation library. This library catalogs pre-reviewed, standardized explanations for common and acceptable issues encountered across projects. The implementation workflow is as follows:

1. Initial Project Interpretation: When the project generates the first P21 report, a special interpretation tool is used to match the questions in the report with the interpretation library and automatically obtain the preset text of the interpretation.
2. Handling Unacceptable or Novel Issues:
    - When unacceptable problems are found, the tool will alert the programmer to modify the corresponding dataset
    - When a new issue is encountered that is not covered by the Interpretation Library, the tool alerts the programmer. (Figure 25):
        - Manually add project-specific explanations to reports.
        - (Optional) If the new question is assessed as common and acceptable, a new iteration of the standardized interpretation can be updated to the central interpretation repository.

WARNING: 模板中暂无解释，需更新模板或手动添加解释 DV SD1075 使用了不推荐使用的变量

Figure 25 Tool Prompt for New Issues

3. Efficient Processing of Subsequent Reports: For follow-up P21 reports within the same project, the interpretation tool automatically retrieves and reuses explanations from the previous report, thus avoiding manual repetitive interpretation operations and ensuring consistency between reports.

6.3 CHALLENGES AND SOLUTIONS IN DRG AUTOMATION

A critical technical challenge in DRG automation is to ensure reliable hyperlink navigation within the document. Our solution leverages the native bookmark-based hyperlinking mechanism of RTF formats (e.g., Figure 26):

- Predefined Common Bookmarks: The automation tool preconfigures bookmarks for frequent cross-referencing needs (e.g., linking "Analysis Dataset Overview" in ADRG to specific dataset sections).
- Flexible Extension Mechanism: To accommodate project-specific hyperlinking needs that go beyond the predefined scope, we introduced an easy-to-use markup syntax in the DRG specification. The programmer simply wraps the name of the target section in curly braces at the point in the specification where the link is needed (e.g.{附录 1：入选/排除标准}). The automated process recognizes this syntax when generating the final document and dynamically creates the corresponding bookmarks and hyperlinks.

```
\trowd\trkeep\trql
\clbrdrb\brdrs\brdrw5\brdrcf1\clbrdrl\brdrs\brdrw2\brdrcf1\cltxlrtb\clvertalt\clcbpat8\clpadt20\clpadft3\clpadr20\clpadfr3\cellx4509
\clbrdrb\brdrs\brdrw5\brdrcf1\clbrdrl\brdrs\brdrw5\brdrcf1\clbrdrr\brdrs\brdrw2\brdrcf1\cltxlrtb\clvertalt\clcbpat8\clpadt20\clpadft3\clpadr20\clpadfr3\cellx9021
\pard\plain\intbl\sb20\sa20\ql\fl\fs21\cf1{{\field {\*\fldinst HYPERLINK \\l IDX6}{\fldrslt\cf2\ul TA}}\cell}
\pard\plain\intbl\sb20\sa20\ql\fl\fs21\cf1{\u35797;\u39564;\u20998;\u32452;\cell}
```

Figure 26. Hyperlinking in RTF code

TI数据集描述了试验的入选/排除标准。变量IETEST（入选/排除标准）概括至长度为200内，详细内容参见{附录1：入选/排除标准}。

Figure 27. TI Dataset Description in CSDRG Spec

TI数据集描述了试验的入选/排除标准。变量IETEST（入选/排除标准）概括至长度为200内，详细内容参见 附录1：入选/排除标准 。

Figure 28 Rendered Output in CSDRG Document

7. PRACTICAL IMPLEMENTATION AND APPLICATION

This process has been successfully applied to Keymed's projects for two years, significantly improving the efficiency and quality of clinical data programming work. Below are the main advantages:

7.1 INTELLIGENT EXPERIENCE: TRANSITION FROM MANUAL TO AUTOMATION

- Full Process Automation: Subvert the traditional manual operation mode and automate key aspects such as aCRF paste, Spec writing, program writing, etc., releasing more than 60% of programmers' repetitive man-hours, and increasing the utilization rate of effective man-hours by 100%.
- Standardized Output: Eliminate more than 50% of human operation differences through the built-in 200+ data processing rules, ensuring highly consistent work results for different projects and members.
- Simplified Code: Reduce code output by 40%, improve logical clarity, and significantly lowers subsequent maintenance and interpretation costs.

7.2 EFFICIENCY IMPROVEMENT: PROJECT CYCLE COMPRESSION BEYOND INDUSTRY STANDARDS

- Batch Processing Capability: Compress repetitive operations (e.g., variable mapping, label generation) that previously took days into minutes, enabling parallel processing of multiple projects.

- Milestone Acceleration (an example of a project with 29 SDTM Domains, 102 pages of aCRFs, and 21 ADaM analysis datasets in the TaiMei EDC system):
    - aCRF Annotation: ≤0.5 days (industry average: 2–3 days)
    - SDTM Full Process Programming: ≤3 days, one person (industry average: 5–7 days, multiple personnel)
    - ADaM Full Process: ≤2 weeks, one person (industry average: 2–3 weeks, multiple personnel)

7.3 QUALITY ENHANCEMENT: DUAL ASSURANCE OF STANDARDIZATION AND ACCURACY

- Built-in Compliance Engine: Integration of the latest CDISC standard model, automatic verification of compliance nodes to ensure compliance with regulatory agency audit requirements.
- Significant Error Reduction: Reduce manual operation links by 50% through automated validation mechanism.

7.4 EASE OF TECHNICAL IMPLEMENTATION

The tool workflow is implemented with SAS as the core (accounting for over 95%), supplemented by lightweight Python and AI modules, significantly reducing technical complexity and deployment barriers. Thanks to the standardized program architecture, the automated SAS code output can be efficiently converted to R language, meeting the needs of diverse analytical environments.

ACKNOWLEDGMENTS

We would like to express our sincere thanks to the development team for their hard work in tool development, and to Keymed for building a good platform for the project. Special thanks to Dr. Guoqing Zhao for his great support in resource coordination, Rongqin Xie for his professional contribution in technical guidance and Mi Li for her valuable suggestions on the paper.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:
Author Name: Fang Wang
Company: Keymed Biosciences Inc.
E-mail: FangWang@keymedbio.com