An Innovative ADaM Programming Tool for FDA Submission

Xiangchen (Bob) Cui, Min Chen
Vertex Pharmaceuticals, Cambridge, MA

ABSTRACT

It is a good practice to include data definition tables (define.xml) and a reviewer's guide along with ADaM datasets to minimize the time to familiarize with submitted clinical data and expedite the approval process by FDA reviewers. It is important to ensure consistency in metadata among data definition tables, reviewer's guide and ADaM datasets. This paper describes automated ADaM Programming Tool, consisting of six SAS macros, to streamline the whole process from creating programming specification to generating a complete ADaM package for FDA electronic submission.

Introduction

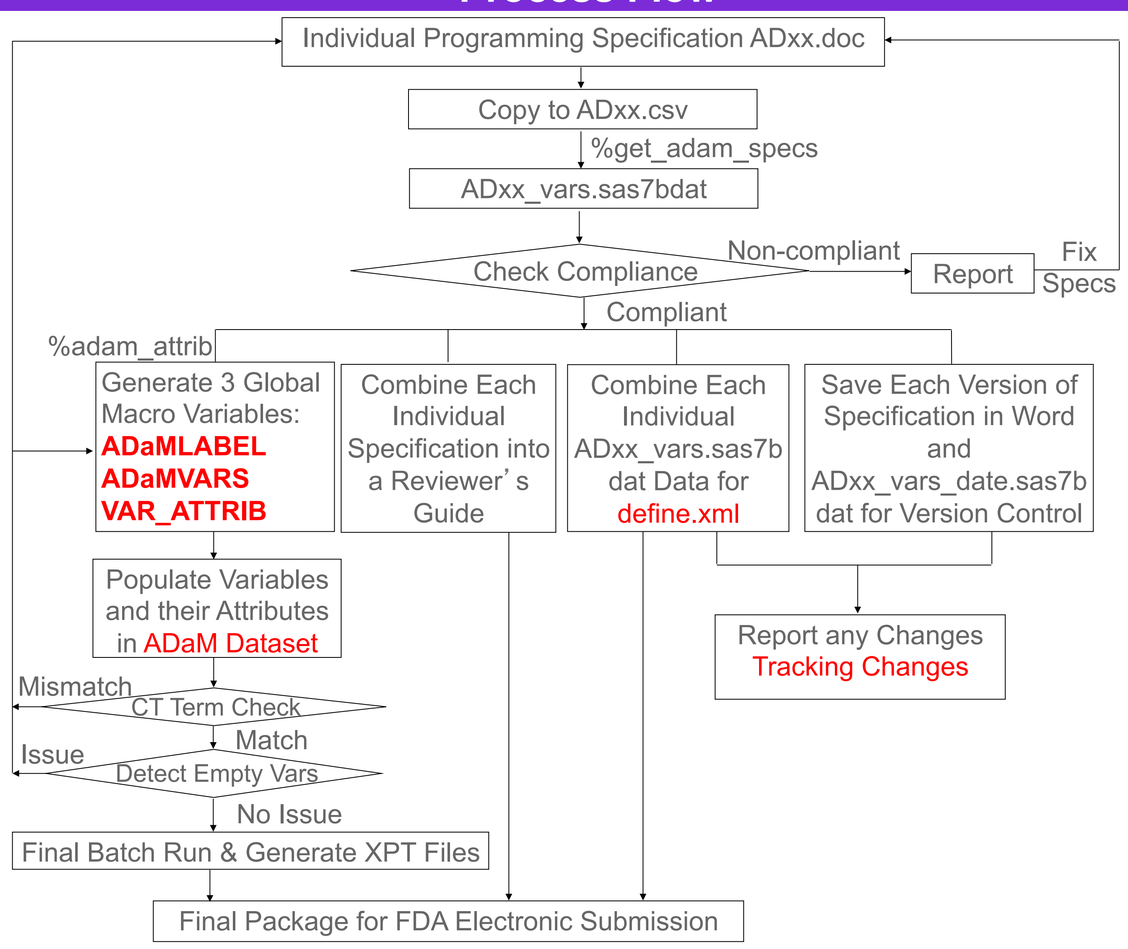
TEN AUTOMATIONS provided by the ADaM Programming Tool:

1. Automatic compliance checking of metadata against CDISC standards
2. Automatic version control
3. Automatic track changes in analysis programming specification
4. Automatic define.xml generation
5. Automatic generation of ADaM dataset label & variable attributes
6. Automatic addition of core variables to both define.xml and ADaM datasets in final run
7. Automatic consistency checking of controlled terms and value level metadata between ADaM datasets & specs
8. Automatic detection of empty variables in ADaM datasets
9. Automatic preparation of SAS scripts for final run of all ADaM specs & combination of all specs into one
10. Automatic batch file preparation for the final run of ADaM datasets

Goal:

- Streamline the Process from ADaM Programming to FDA eSub
- Meet the Requirements in ADaM Implementation Guide and In-house Standards
- Guarantee Consistency Among ADaM Datasets, Programming Specification, and define files
- Enhance the Submission Quality
- Achieve Cost-Effectiveness and Efficiency

Process Flow



Function of Innovative ADaM Tool

1.1.1 ADSL: Subject Level Analysis Dataset — Domain Information Table

Dataset	**ADSL**
Program Name	Adsl.sas
Description	Subject-Level Analysis Data
Unique identifier Variables	usubjid
General Class	Special Purpose
Structure	One record per subject
Input Datasets	DM, DS, EX, VS, DC, HC, AE
Notes	Includes all subjects enrolled.

Compliant →

Variable Name	Variable Label	Type	Length	Controlled Terms or Formats	Origin	Role	Comments	Core
STUDYID	Study Identifier	Char	20		DM.studyid	Identifier	Constant Value: "ABC-ZZZ-XXX"	Req
USUBJID	Unique Subject Identifier	Char	40		DM.usubjid	Identifier	Equivalent to studyid || "-" || strip(siteid) || "-" || strip(subid)	Req
SUBJID	Subject Identifier for the Study	Char	20		DM.subjid	Identifier	(e.g. 102130)	Req
SITEID	Study Site Identifier	Char	8		DM.siteid	Record Qualifier	DM.SITEID	Req
AGE	Age	Num	8		DM.age	Record Qualifier	Equals to DM.age	Req
AGEGR1	Pooled Age Group 1	Char	20		Derived	Record Qualifier	<=45, if age <= 45 >45 and <=65, if 45 < age <= 65 >65, if age > 65 Note: Decode variable for AGEGPN.	Perm
AGEGR1N	Pooled Age Group 1 (N)	Num	8	AGEGR1N (AGEGR1): (1) 1 = <=45 (2) 2 = >45 and <=65 (3) 3 = >65	Derived	Synonym Qualifier	Category derived if age non-missing. Equals 1, if age <= 45 2, if 45 < age <= 65 3, if age > 65	Perm

Controlled Terms or Value Level Metadata

Variable Information Table

Non-Compliant ↓

Typical Non-Compliance Reports

The Following Variables with Non-compliance Specs.!

Variable	Error Type
O_TIMDIAG	VAR Name > 8 Chars

Output (SAS Datasets)

Individual ADaM Data Information
- ADXX_DOMAIN
- ADXX_VARS:

↓

Generate 3 global macro variables: &adamlabel, &adamvars, &var_attrib inside ADaM derivation program for dataset label and variable attributes

All ADaM Data Information
- ALL_DOMAINS
- ALL_VARS

↓

- Generate define.xml
- Add core variables to define.xml & ADaM data
- Generating 3 files for final run:
  - ➢a SAS program for macro calls of %get_adam_specs
  - ➢a SAS program combining all ADaM specs into ONE for reviewer guide
  - ➢a BAT file for final run

Other Function of Innovative ADaM Tool

Typical Report of Tracking Changes

The following Variables with Comment Changed!

Variable	Variable Nmber	Source	Comment
AGE	5	New Specs	Equals to DM.age
		Old Specs	age = (BRTHDTC-RFSTDTC)/365.25

Typical Report of Empty Variables

The Following ADaM Variables in Study xxx Have All Missing Values

Domain	Variable Order	Variable	Variable Label	Total # of Observations	Core	Comment
ADAE	73	DCRASHFL	Discontinuation due to Rash SSC	561	Perm	Warning: Permissible Variable is Empty. Delete it?
	74	DCPRURFL	Discontinuation due to Pruritus SSC	561	Perm	Warning: Permissible Variable is Empty. Delete it?

Typical Report for Consistency Checking of Controlled Terms and Value Level Metadata

The following PARCAT1 Variables with Different Value Level Metadata between Programming Specs. and Datasets

Domain	Variable	Value	Value Label in Dataset	Value Label in Specs.	Terms In Specs. NOT in Dataset	Terms In Dataset NOT In Specs.
ADLB	PARCAT1	VIROLOGY	VIROLOGY			Yes

The following PARAM Variables with Different Value Level Metadata between Programming Specs. and Datasets

Domain	Variable	Variable Label	Value	Value Label in Dataset	Value Label in Specs.	Terms In Specs. NOT in Dataset	Terms In Dataset NOT In Specs.	Different Controlled Terminology
ADVS	PARAMCD	Parameter Code	RESP	RESPIRATORY RATE (BREATHS/MIN)			Yes	
			BMI	BODY MASS INDEX (KG/M2)	BODY MASS INDEX			Yes
			RESPR		RESPIRATORY RATE (BREATHS/MIN)	Yes		

The following Coded Variables with Different Decoded Terminology between Programming Specs. and Datasets

Domain	Variable	Variable Label	Coded Controlled Term	Decoded Controlled Term in Dataset	Decoded Controlled Term in Specs.	Codelist in Specs.	Terms In Specs. NOT in Dataset	Terms In Dataset NOT In Specs.	Different Decoded Terminology
ADAE	AEACNN	Action with Study Treatment Number	5	NOT APPLICABLE		AEACNN		Yes	
	AEOUTN	Outcome of Adverse Event Number	3	RECOVERED/ RESOLVED WITH SEQUELAE	RECOVERED/RESOLVED WITH SEQUELAE	AEOUTN			Yes
			4		FATAL	AEOUTN	Yes		

Decision Making On the Mismatches Between ADaM Specs and Datasets

#	Scenario	Condition	Action Taken
1	Controlled Terms or Value Lists are not in the Datasets but in the Specs	Code lists are correctly defined in specs	No Action Needed
2	Controlled Terms or Value Lists are in the Datasets but not in the Specs	Specification does not list all the possible values for the controlled terms or value lists	Add Missing Controlled Terms/Value Lists to specs
3	Code Value for Sponsor-Defined Controlled Terminology or Value for PARAMCD are Differently Defined in the Datasets from that in the Specs	Code Value or Value in datasets is not consistent with Standard Controlled Terms	Revise ADaM Datasets
		Code Value or Value in specs is not consistent with Standard Controlled Terms	Revise ADaM Specs
4	Decoded Value for Sponsor-Defined Controlled Terminology or Value Label for PARAMCD are Differently Defined in the Datasets from that in the Specs	Decode Value or Value Label in datasets is not consistent with Standard Controlled Terms	Revise ADaM Datasets
		Decode Value or Value Label in specs is not consistent with Standard Controlled Terms	Revise ADaM Specs
5	Typo		Correct the typo

Comparison of Conventional Methodology and New One

Comparison	Conventional Methodology	New Methodology
1. Automation of **Compliance Checking** CDSIC ADAM PROGRAMMING SPECIFICATION	No	• **DETECT ANY NONCOMPLIANCE** with FDA submission requirements, **CDISC ADaM GUIDELINE,** & **Vertex Guideline** for writing specs • **REPORT** any findings • **ENSURE** the resolution • **FINALIZE** Specs. earlier in programming cycle
2. Version Control of Programming Specs	No	• Keep both word version and SAS dataset of programming specs in a study subfolder \History
3. Track Changes of Programming Specification	No	• Report changes from any previous version of specs • Help developer & reviewers to trace back changes • Serve as a tool for audit from post delivery changes
4. define.xml Generation	• Timing: 2 stages (First CSR, second define.xml) • Trainings needed • Additional Resources/ Times to prepare spreadsheets	• Same time as CSR A tool for statisticians to review metadata earlier • Minimal Training Needed • No Extra Resources/Times Automatic generation per the macro user request
5. Generation of Variable Attributes Any Changes of Variable & Attributes and Adding/ Deleting Variables	• Write an additional SAS program to generate a template • Insert it into data step in SAS • Update both SAS template program and specs • More time needed • Error-prone, inconsistent	• Call a SAS macro to automatically generate a dataset by extracting from the programming specs. • Call a SAS macro to automatically generate SAS Macro Variables to be used in final data step of SAS program • Update Programming Specifications ONLY • Less time needed, cost-effective • Ensure quality & consistency between data & specs
6. Adding Core Variables into ADaM Data and define.xml for Final Run	• Develop additional SAS programs to populate core variables into ADaM Data and define.xml • Error-prone & labor-intensive!	• Automatically generate define.xml with core variables • Automatically generate a SAS program for populating core variables into ADaM Datasets • Automation saves time and energy!
7. Automatic Consistency Checking of Controlled Terms between ADaM data and specs	No	• **DETECT ANY MISMATCHES** between ADaM Datasets and specs. for Controlled Terminology and Value Level Metadata **at any stage** of programming cycle • **REPORT** any findings • **FINALIZE** Specs. earlier in programming cycle
8. Automatic Detection of Empty Variables in ADaM Datasets	No	• **DETECT ANY EMPTY VARIABLES** • **REPORT** any findings with the information of core attributes • **DECISION MAKING AND FINALIZE** Specs **at early stage**
9. Combination of All ADaM Specs. into one	• Manually copy and paste	• Automatically Combine All Individual ADaM Specs. into ONE Word Document
10. Batch File Preparation for Final Run of ADaM Datasets	• Error-prone Type-In or Copy/Paste	• Automatic Generation with High Quality

Conclusion

➢ADaM dataset structure and define.xml are all generated from the programming specification in Word, the methodology ensures the consistency & high quality of submission.

➢The tool performs compliance checking & consistency checking of controlled terminology and value level metadata at early stage, & avoids unnecessary repetitive work at later stage.

➢The tool can generate define.xml at any stage to facilitate statisticians to review metadata.

➢The tool achieves the cost-effectiveness and the efficiency.

➢The tool is easy to use and only needs minimal training.

Acknowledgement

Appreciation goes to Ana, Tathabbai, Jiannan, and Jason for their review and comments. Special thanks to Stacy for her supporting.

Disclosure

Authors of this presentation have the following to disclose concerning possible financial or personal relationships with commercial entities that may have a direct or indirect interest in the subject matter of this presentation:
Xiangchen (Bob) Cui, Min Chen: Employees – Vertex Pharmaceuticals