Development of ADaM creation tool Towards future Automation

Ryo Nakaya, Fumihiro Yamasaki, Yohei Takanami
Takeda Pharmaceutical Company
@PharmaSUG SDE-Japan

Better Health, Brighter Future

Background

Automation in the creation of statistical deliverables such as SDTM, ADaM, TFLs is the key to the development of the streamlined business process in drug development to reduce cost, save time, and maximize the quality and productivity. Recently many attempts can be seen in the industry.

- Automated generation tool
- low-code and hyper-automation solutions

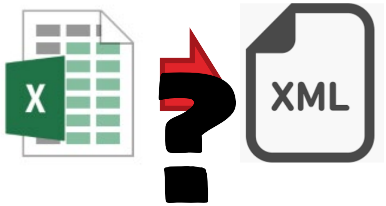
We introduce one of our attempts to develop a tool that generates ADaM datasets with a standardized and streamlined process towards future automation as part of our business process internalization efforts, which has resulted in a reduced workload, time and cost while keeping high quality.

Motivation of Developing ADaM Creation Tool

Challenges in Outsourced Model

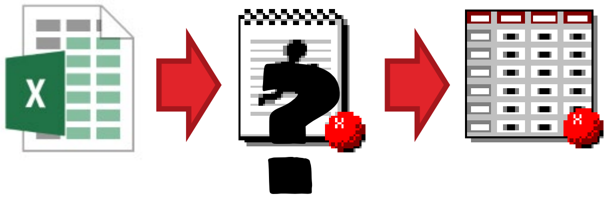
Specs

- <u>Different spec formats</u> by several vendors
- Unclear Define.xml creation process
  (no ad-hoc reproductivity)
- Takeda Guidance of ADaM was created with general guidance(more review needed in studies) <u>without machine readability for automation</u>



Programs

- Manually writing programs
  - <u>Inefficient</u>
  - <u>Error-prone</u>
- <u>Black box</u>, that is, Difficulty in modifying and re-running programs without macros of vendors' intellectual property during e-Data preparation and submission process



Transition from Outsource model to In-house model

Needs to have templates/tools in order to <u>**create statistical deliverables in-house**</u> as well as <u>**to better manage cases of outsourced models**</u>.

Agenda

- Statistical Deliverables and Automation in Clinical Development
- Semi-automated ADaM Creation Tool in Takeda
- Use cases, Lessons Learned and Challenges
- Summary and Next Step

Agenda

- Statistical Deliverables and Automation in Clinical Development
- Semi-automated ADaM Creation Tool in Takeda
- Use cases, Lessons Learned and Challenges
- Summary and Next Step

Statistical Deliverables and Automation in Clinical Development

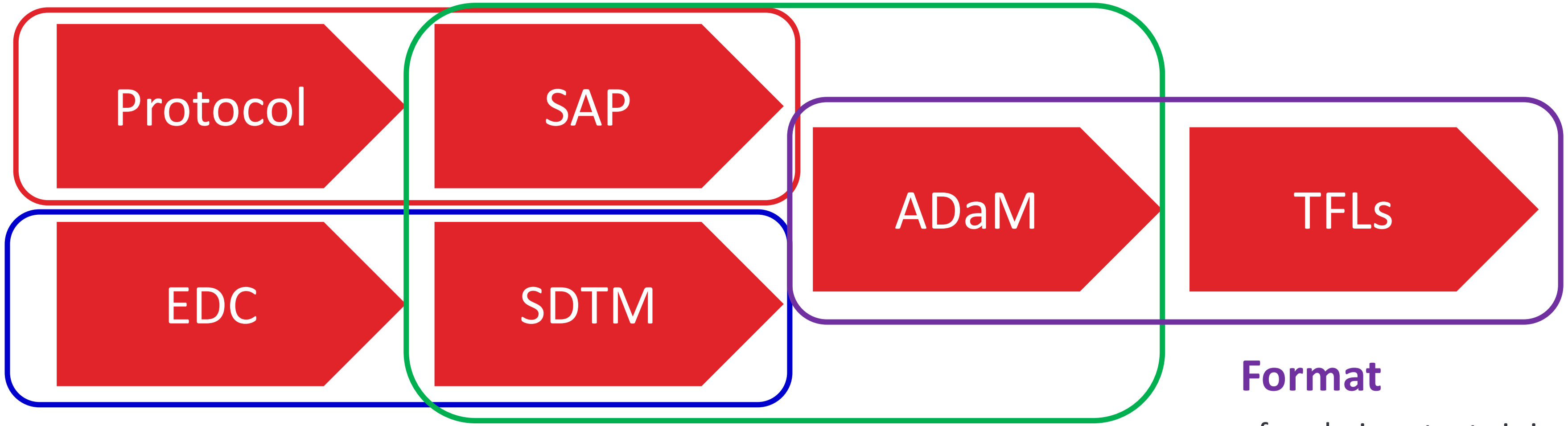
Today's Scope

Analysis document

created based on protocol. Summaries of major analysis plan in protocol and further details need to be added in SAP. A consideration on how much complete or machine readable for ADaM/TFL creation is necessary.

Creation of analysis datasets

using SAP and source SDTM datasets. There are common datasets across studies and therapeutic areas while additional datasets specific to study/therapeutic area.



Format

of analysis outputs is important. Using Standardized TFL shell can bring about highly re-usable programs/macros and reduce programming workload.

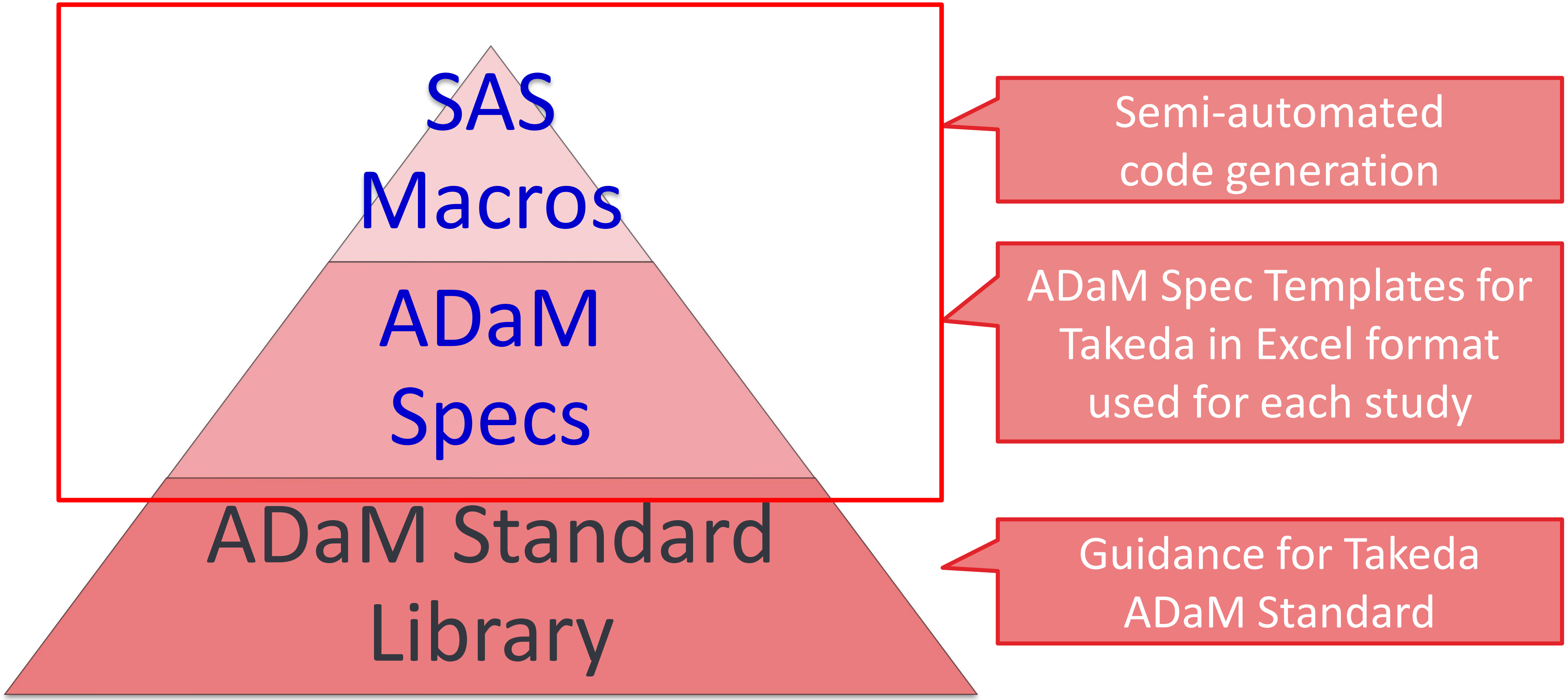
Mapping

from various source data including EDC, lab, other external data is needed. Less derivation than ADaM creation.

Agenda

- Statistical Deliverables and Automation in Clinical Development
- Semi-automated ADaM Creation Tool in Takeda
- Use cases, Lessons Learned and Challenges
- Summary and Next Step

Semi-automated ADaM Creation Tool in Takeda



Semi-automated code generation

ADaM Spec Templates for Takeda in Excel format used for each study

Guidance for Takeda ADaM Standard

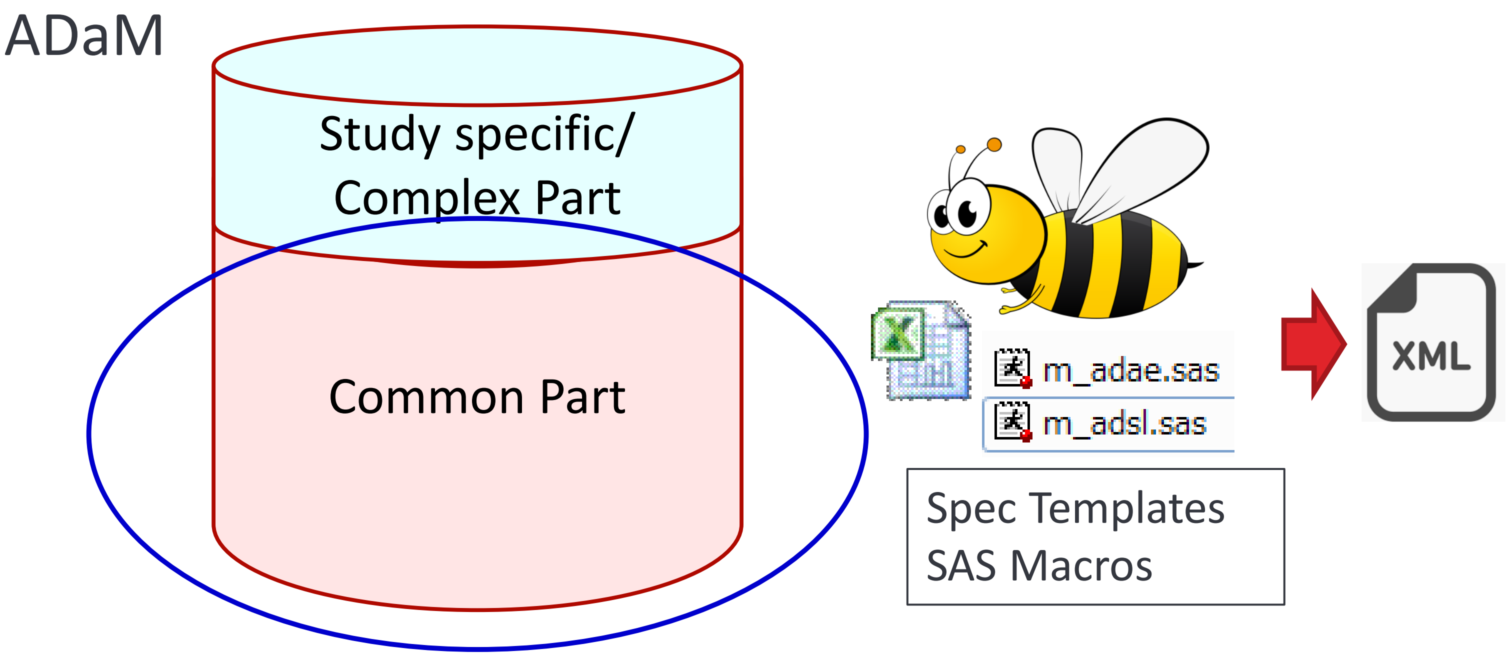
Semi-automated ADaM Creation Tool in Takeda

The tool

- Consists of ADaM Spec Templates and SAS Macros
- Has Excel formatted templates with default values for Common(Base) part of ADaM
- Has macros that automatically generate SAS codes by reading the spec
- Generates define.xml using **tsClinical Define.xml Generator©*

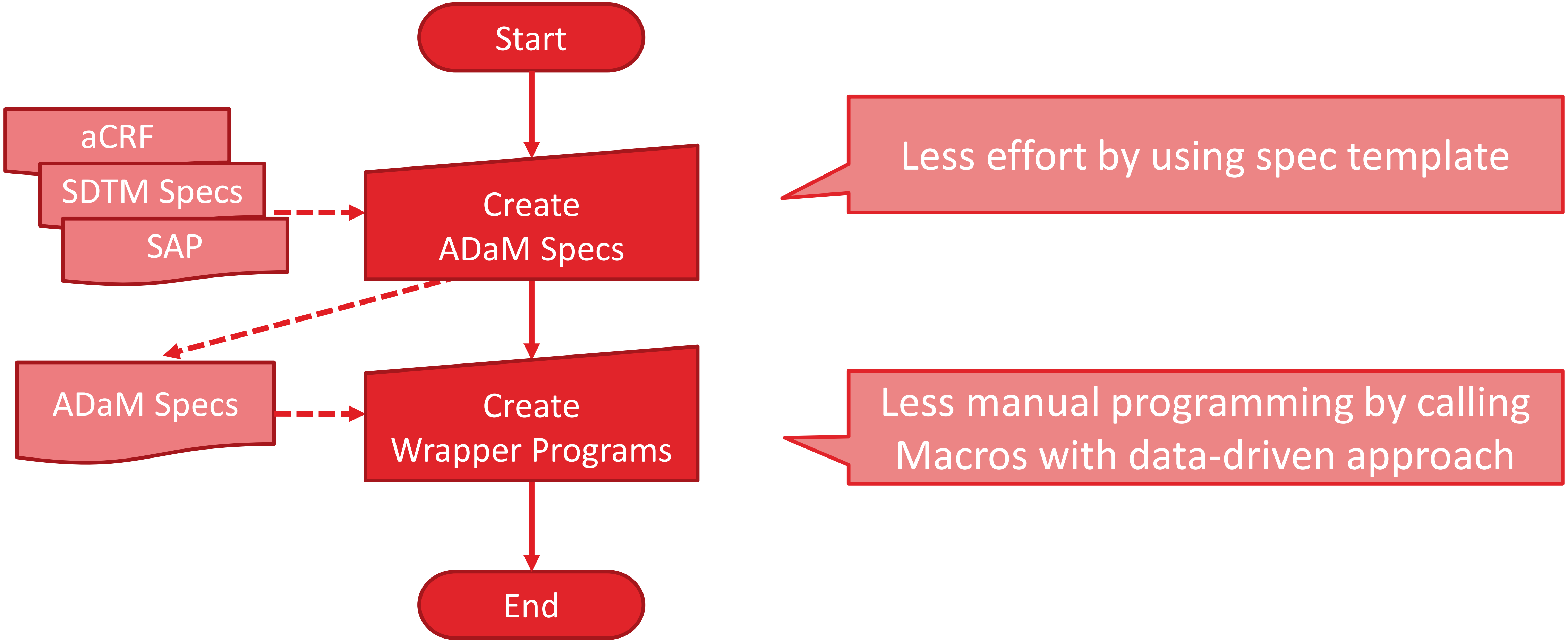
*product of Fujitsu

As of Oct2023, the tool is available as part of tsClinical Metadata Desktop Tools. https://github.com/tsClinical/tsc-desktop



Spec Templates
SAS Macros

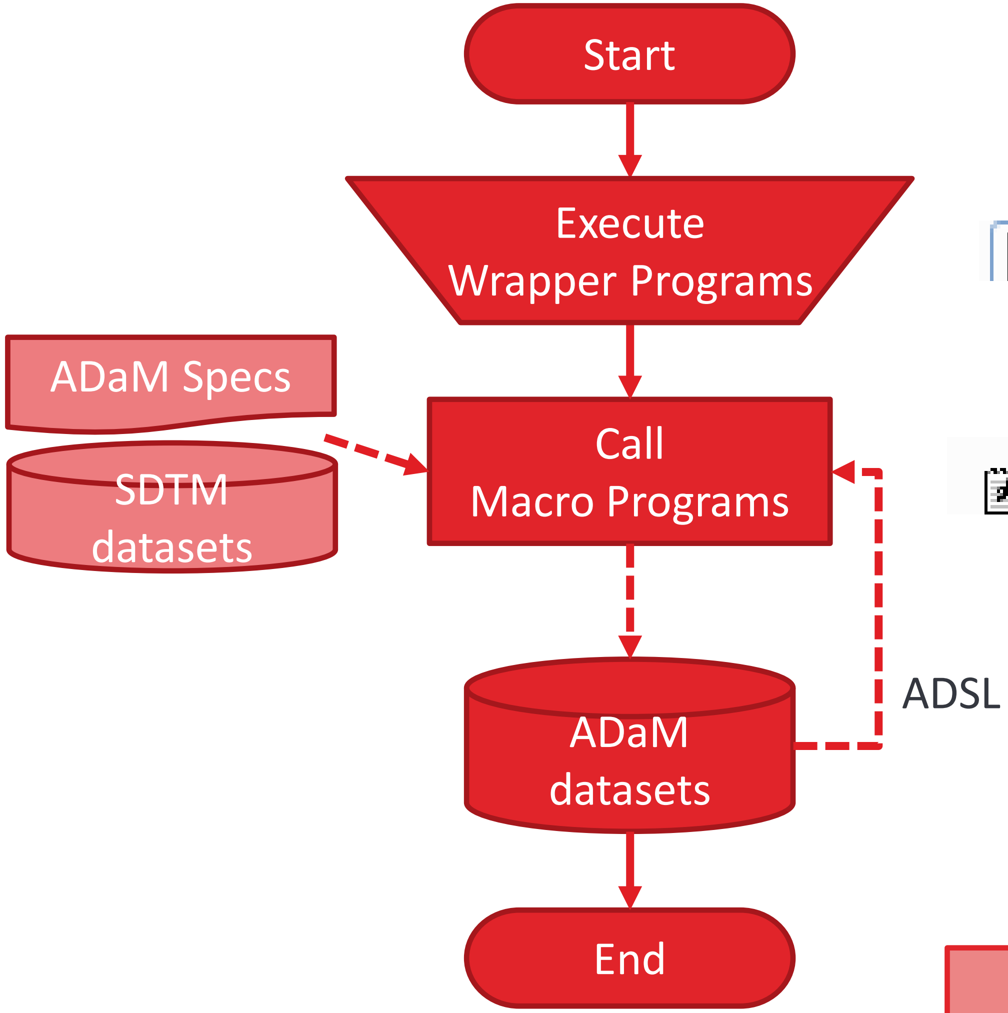
Process flow of Creating Spec and Programs

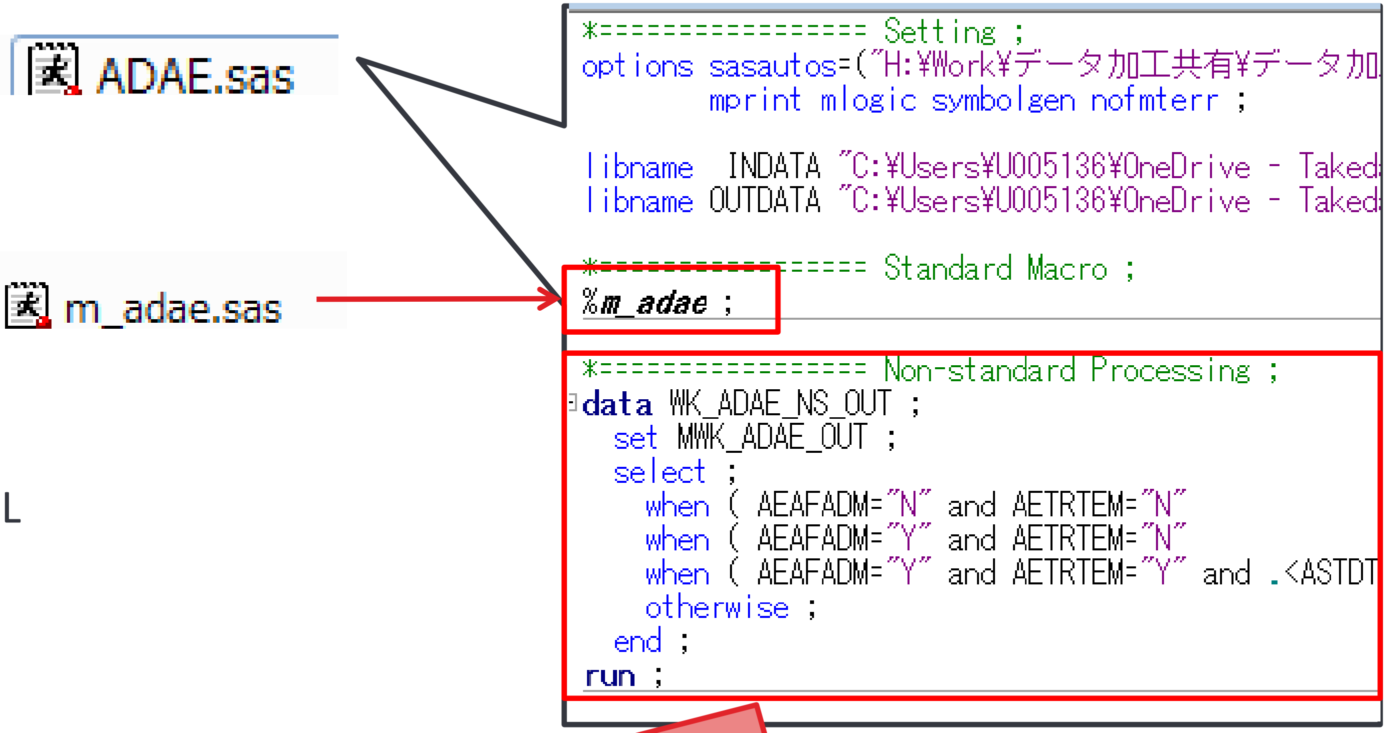


Less effort by using spec template

Less manual programming by calling Macros with data-driven approach

Process flow of Executing Programs to create Datasets





```
*================= Setting ;
options sasautos=("H:¥Work¥データ加工共有¥データ加
        mprint mlogic symbolgen nofmterr ;

libname  INDATA "C:¥Users¥U005136¥OneDrive - Taked
libname OUTDATA "C:¥Users¥U005136¥OneDrive - Taked

*================= Standard Macro ;
%m_adae ;

*================= Non-standard Processing ;
data WK_ADAE_NS_OUT ;
  set MWK_ADAE_OUT ;
  select ;
    when ( AEAFADM="N" and AETRTEM="N"
    when ( AEAFADM="Y" and AETRTEM="N"
    when ( AEAFADM="Y" and AETRTEM="Y" and .<ASTDT
    otherwise ;
  end ;
run ;
```

Study/therapeutic specific additional coding.

Spec Templates

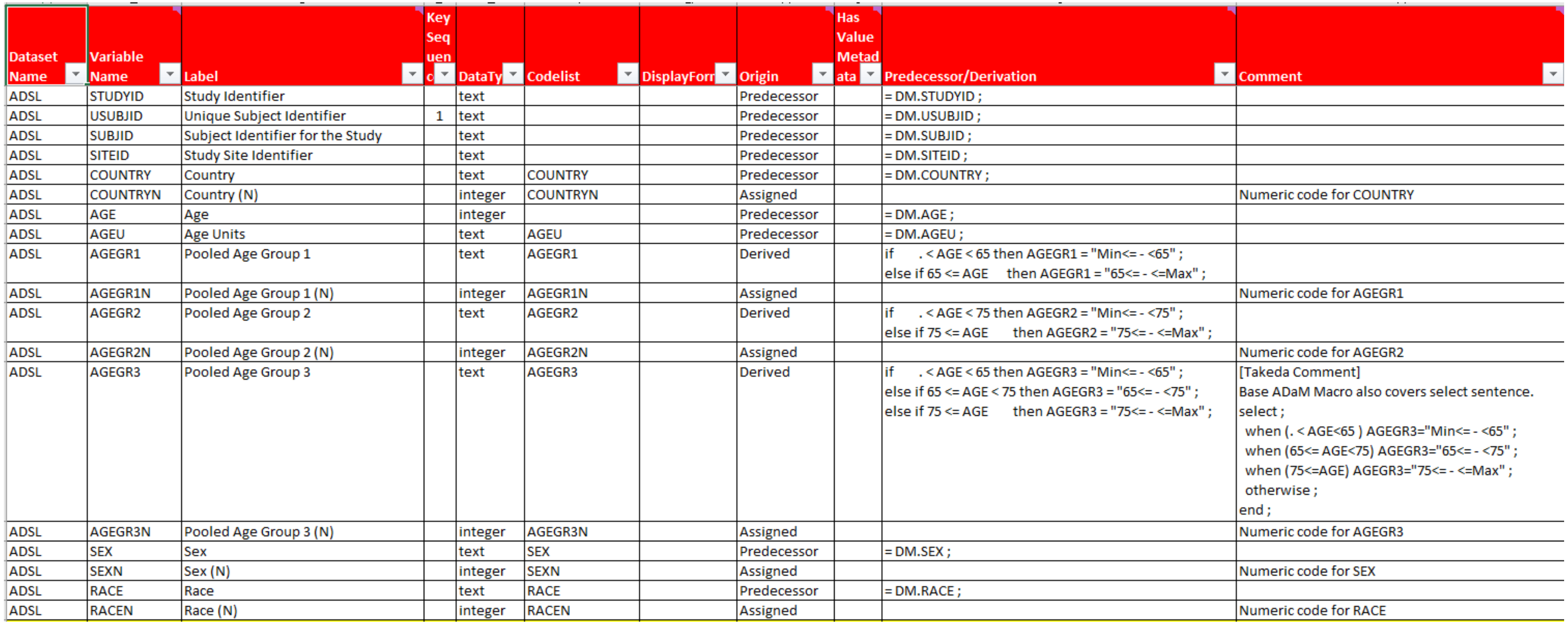
tsClinical Define.xml Generator© based spec. Moved and added some columns and sheets for efficiency in creation and review.

Cover common datasets

- ADSL
- ADAE, ADCM, ADMH, ADDV (OCCDS)
- ADLB, ADVS, ADEG, ADPC, ADPP (BDS)

Users to modify sample logic(only if needed) and add study specific variables.

Cover common variables / common derivation logic (sample logic)



Dataset Name	Variable Name	Label	Key Sequenc	DataTy	Codelist	DisplayFor	Origin	Has Value Metadata	Predecessor/Derivation	Comment
ADSL	STUDYID	Study Identifier		text			Predecessor		= DM.STUDYID ;	
ADSL	USUBJID	Unique Subject Identifier	1	text			Predecessor		= DM.USUBJID ;	
ADSL	SUBJID	Subject Identifier for the Study		text			Predecessor		= DM.SUBJID ;	
ADSL	SITEID	Study Site Identifier		text			Predecessor		= DM.SITEID ;	
ADSL	COUNTRY	Country		text	COUNTRY		Predecessor		= DM.COUNTRY ;	
ADSL	COUNTRYN	Country (N)		integer	COUNTRYN		Assigned			Numeric code for COUNTRY
ADSL	AGE	Age		integer			Predecessor		= DM.AGE ;	
ADSL	AGEU	Age Units		text	AGEU		Predecessor		= DM.AGEU ;	

ADSL	AGEGR1	Pooled Age Group 1		text	AGEGR1		Derived		if . < AGE < 65 then AGEGR1 = "Min<= - <65" ;
									else if 65 <= AGE then AGEGR1 = "65<= - <=Max" ;	

ADSL	AGEGR1N	Pooled Age Group 1 (N)		integer	AGEGR1N		Assigned			Numeric code for AGEGR1

ADSL	AGEGR2	Pooled Age Group 2		text	AGEGR2		Derived		if . < AGE < 75 then AGEGR2 = "Min<= - <75" ;
									else if 75 <= AGE then AGEGR2 = "75<= - <=Max" ;	

ADSL	AGEGR2N	Pooled Age Group 2 (N)		integer	AGEGR2N		Assigned			Numeric code for AGEGR2

ADSL	AGEGR3	Pooled Age Group 3		text	AGEGR3		Derived		if . < AGE < 65 then AGEGR3 = "Min<= - <65" ;
									else if 65 <= AGE < 75 then AGEGR3 = "65<= - <75" ;
									else if 75 <= AGE then AGEGR3 = "75<= - <=Max" ;
										[Takeda Comment]
										Base ADaM Macro also covers select sentence.
										select ;
										when (. < AGE<65) AGEGR3="Min<= - <65" ;
										when (65<= AGE<75) AGEGR3="65<= - <75" ;
										when (75<=AGE) AGEGR3="75<= - <=Max" ;
										otherwise ;
										end ;

ADSL	AGEGR3N	Pooled Age Group 3 (N)		integer	AGEGR3N		Assigned			Numeric code for AGEGR3
ADSL	SEX	Sex		text	SEX		Predecessor		= DM.SEX ;	
ADSL	SEXN	Sex (N)		integer	SEXN		Assigned			Numeric code for SEX
ADSL	RACE	Race		text	RACE		Predecessor		= DM.RACE ;	
ADSL	RACEN	Race (N)		integer	RACEN		Assigned			Numeric code for RACE

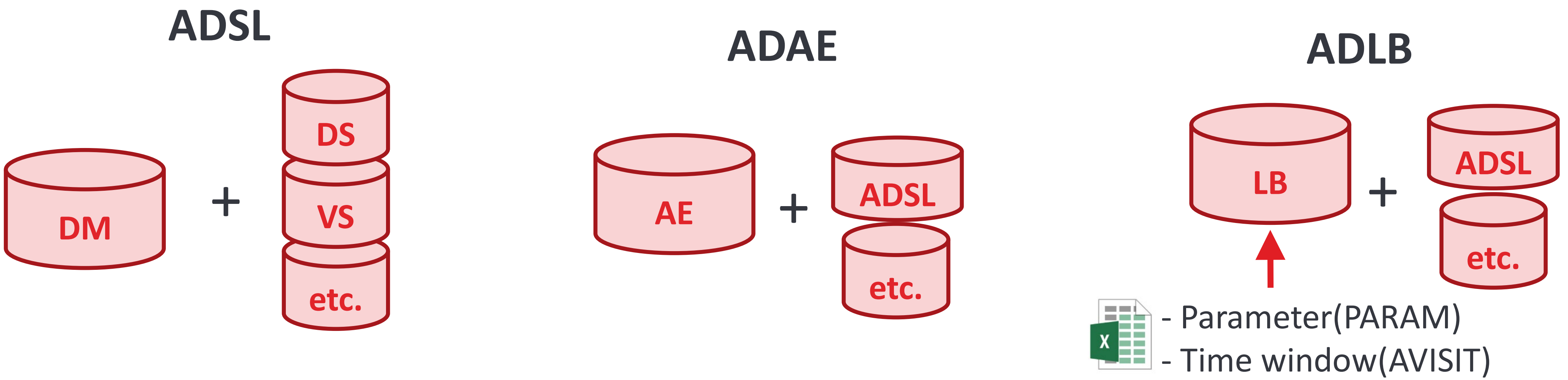
SAS Macros

Cover common datasets

- ADSL
- ADAE, ADCM, ADMH, ADDV (OCCDS)
- ADLB, ADVS, ADEG (BDS)

Features and functionality of Macros

- Dataset-level core macros consisting of small functional macros with better maintainability
- <u>Framework of ADaM datasets is to be built</u> by the core macros (incl. parameter, time window)
- <u>Metadata-driven code creation is used for common derivations</u>

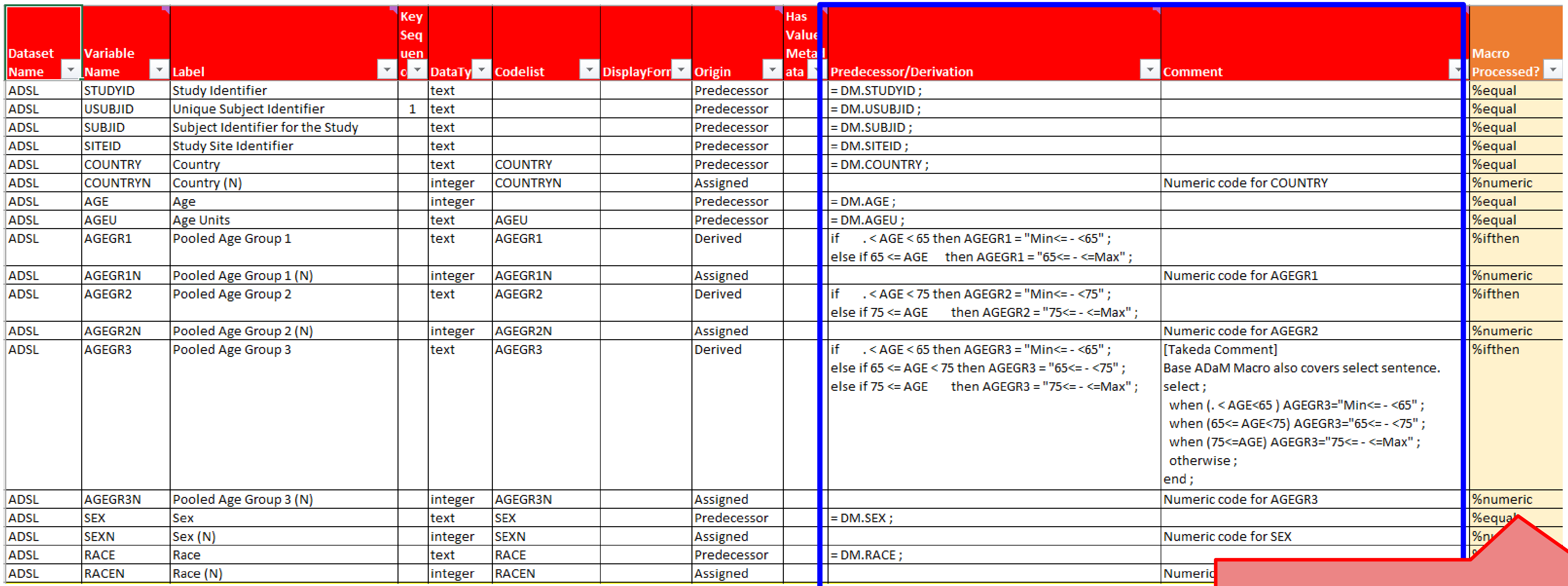


- Parameter(PARAM)
- Time window(AVISIT)

Metadata-driven Code Creation for Common Derivations

Describe derivation in pre-defined rules to utilize metadata-driven functionality

Core macros read spec file and generate SAS codes



Dataset Name	Variable Name	Label	Key Seq uen c	DataTy	Codelist	DisplayForr	Origin	Has Value Meta ata	Predecessor/Derivation	Comment	Macro Processed?
ADSL	STUDYID	Study Identifier		text			Predecessor		= DM.STUDYID ;		%equal
ADSL	USUBJID	Unique Subject Identifier	1	text			Predecessor		= DM.USUBJID ;		%equal
ADSL	SUBJID	Subject Identifier for the Study		text			Predecessor		= DM.SUBJID ;		%equal
ADSL	SITEID	Study Site Identifier		text			Predecessor		= DM.SITEID ;		%equal
ADSL	COUNTRY	Country		text	COUNTRY		Predecessor		= DM.COUNTRY ;		%equal
ADSL	COUNTRYN	Country (N)		integer	COUNTRYN		Assigned			Numeric code for COUNTRY	%numeric
ADSL	AGE	Age		integer			Predecessor		= DM.AGE ;		%equal
ADSL	AGEU	Age Units		text	AGEU		Predecessor		= DM.AGEU ;		%equal

ADSL	AGEGR1	Pooled Age Group 1		text	AGEGR1		Derived		if . < AGE < 65 then AGEGR1 = "Min<= - <65" ;
									else if 65 <= AGE then AGEGR1 = "65<= - <=Max" ;
											%ifthen

ADSL	AGEGR1N	Pooled Age Group 1 (N)		integer	AGEGR1N		Assigned			Numeric code for AGEGR1	%numeric

ADSL	AGEGR2	Pooled Age Group 2		text	AGEGR2		Derived		if . < AGE < 75 then AGEGR2 = "Min<= - <75" ;
									else if 75 <= AGE then AGEGR2 = "75<= - <=Max" ;
											%ifthen

ADSL	AGEGR2N	Pooled Age Group 2 (N)		integer	AGEGR2N		Assigned			Numeric code for AGEGR2	%numeric

ADSL	AGEGR3	Pooled Age Group 3		text	AGEGR3		Derived		if . < AGE < 65 then AGEGR3 = "Min<= - <65" ;
									else if 65 <= AGE < 75 then AGEGR3 = "65<= - <75" ;
									else if 75 <= AGE then AGEGR3 = "75<= - <=Max" ;
										[Takeda Comment]
										Base ADaM Macro also covers select sentence.
										select ;
										when (. < AGE<65) AGEGR3="Min<= - <65" ;
										when (65<= AGE<75) AGEGR3="65<= - <75" ;
										when (75<=AGE) AGEGR3="75<= - <=Max" ;
										otherwise ;
										end ;
											%ifthen

ADSL	AGEGR3N	Pooled Age Group 3 (N)		integer	AGEGR3N		Assigned			Numeric code for AGEGR3	%numeric
ADSL	SEX	Sex		text	SEX		Predecessor		= DM.SEX ;		%equa
ADSL	SEXN	Sex (N)		integer	SEXN		Assigned			Numeric code for SEX	%n
ADSL	RACE	Race		text	RACE		Predecessor		= DM.RACE ;		%
ADSL	RACEN	Race (N)		integer	RACEN		Assigned			Numeric	

A column to show variables to be processed by macro

Example of derivations

Syntax	Description	Examples	Generated Codes

= *Dataset-name*.*Variable-name* **;**	Assign values of the specified dataset and variable	= DM.AGE ;	**data** ADSL ;
			  merge ADSL
			          DM(keep=USUBJID AGE rename=(AGE = _AGE)) ;
			  by USUBJID ;
			  AGE = _AGE ;
			run ;

= *Dataset-name*.*Variable-name* **where** *Where-expression* **;**	Assign values of the specified dataset and variable after narrowing down to *one-record-per-subject* using extraction condition in *where statement*	= VS.VSSTRESN where VS.VSTESTCD = "HEIGHT" ;	**data** ADSL ;
			  merge ADSL
			          VS(keep=USUBJID VSTESTCD VSSTRESN
			              rename=(VSTESTCD = _VSTESTCD
			                          VSSTRESN = _VSSTRESN)
			              where=(_VSTESTCD = “HEIGHT”)) ;
			  by USUBJID ;
			  HTBL = _VSSTRESN ;
			run ;

If *Condition1* **then** *Result1* **;**
else if *Condition2* **then** *Result2* **;**
else *Result3* **;**
	Assign values using if/then statement. Select/when can be applied as well.	if . < WTBL < 50 then WTBLGR1 = "Min<= - <50" ; else if 50 <= WTBL then WTBLGR1 = “50<= - <=Max" ;	**data** ADSL ;
			  set ADSL ;
			  if . <= WTBL < 50 then WTBLGR1 = ‘Min<= - <50’ ;
			  else if 50 <= WTBL then WTBLGR1 = '50<= - <=Max’ ;
			run ;

Numeric code for *Variable-name*	Assign numeric code using codelist information for character variable	Numeric code for WTBLGR1	**data** ADSL ; set ADSL ; WTBLGR1N = input(WTBLGR1, WTBLGR1N.) ; **run** ; * WTBLGR1N format was created using CODELIST sheet in advance.

Agenda

- Statistical Deliverables and Automation in Clinical Development
- Semi-automated ADaM Creation Tool in Takeda
- Use cases, Lessons Learned and Challenges
- Summary and Next Step

Use cases, Lessons Learned and Challenges

Experiences in use cases

- Spec templates cover common datasets/variables
- Macros cover coding of **50-80%** in ADSL, **70-90%** in BDS variables, **90% or more** in OCCDS

Learned cases tend to have smaller coverage

- External file for derivation
- Cyclic time window (especially in oncology)

	ADSL	ADAE	ADLB
Study AAA	45%	90%	90%
Study BBB	75%	92%	67%
Study CCC	82%	92%	80%
Study DDD	77%	100%	79%
Study EEE	87%	100%	96%

*cover rate is based on how many of variables did not require manual coding.

Challenges

- Pinnacle format of spec is commonly used in global region
- **CSV(Computerized System Validation)** to be discussed
- **Room to utilize generative AI tools** in drafting study spec and code generation in macros (rule-based derivation can be more flexible)

Agenda

- Statistical Deliverables and Automation in Clinical Development
- Semi-automated ADaM Creation Tool in Takeda
- Use cases, Lessons Learned and Challenges
- Summary and Next Step

Summary and Next Step

With the industry standard, innovative technologies such as CDISC and AI/ML, the business process in the creation of statistical deliverables can be simplified and streamlined towards a fully automated process or system.

More than 80% of ADaM datasets/variables can be automatically generated using the ADaM creation tools and enhancement and improvement are underway.

Technology-based innovation needs a variety of knowledge to realize and keywords to achieve the innovation would be *"Communication"* and *"Collaboration"*.

Better Health, Brighter Future