



doLoop Technologies

Business Analytics. Delivered.

Combining Process Metadata and CDISC Metadata to Achieve Automation

By – Mousum Dutta & Dr.Sangram Parbhane

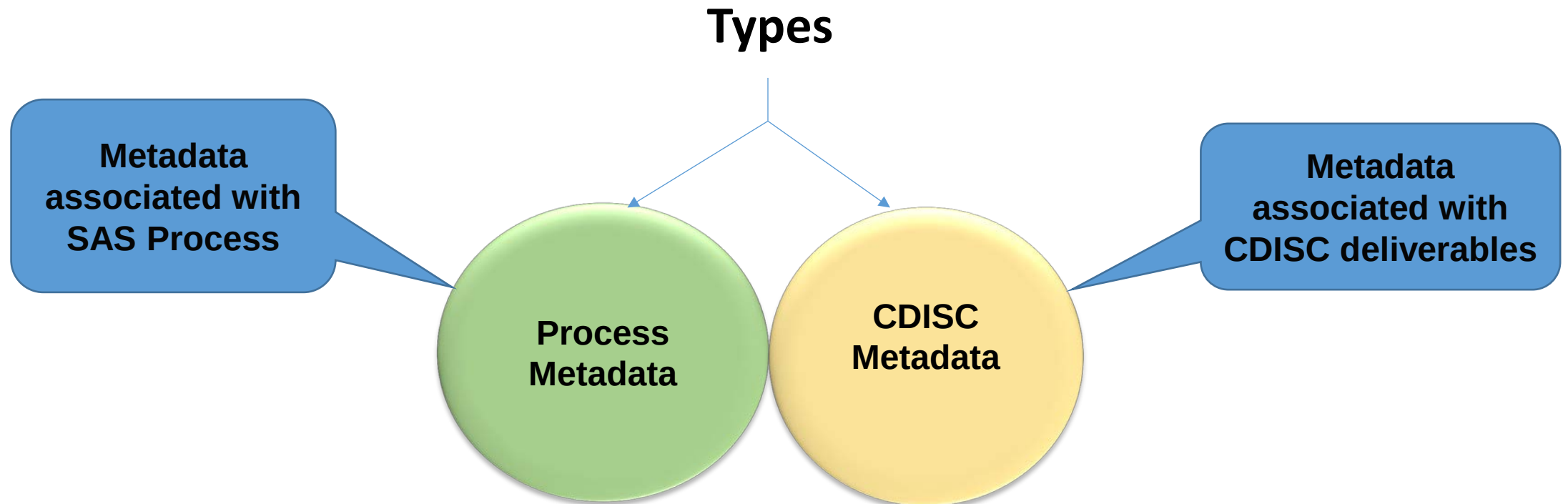
Disclaimers

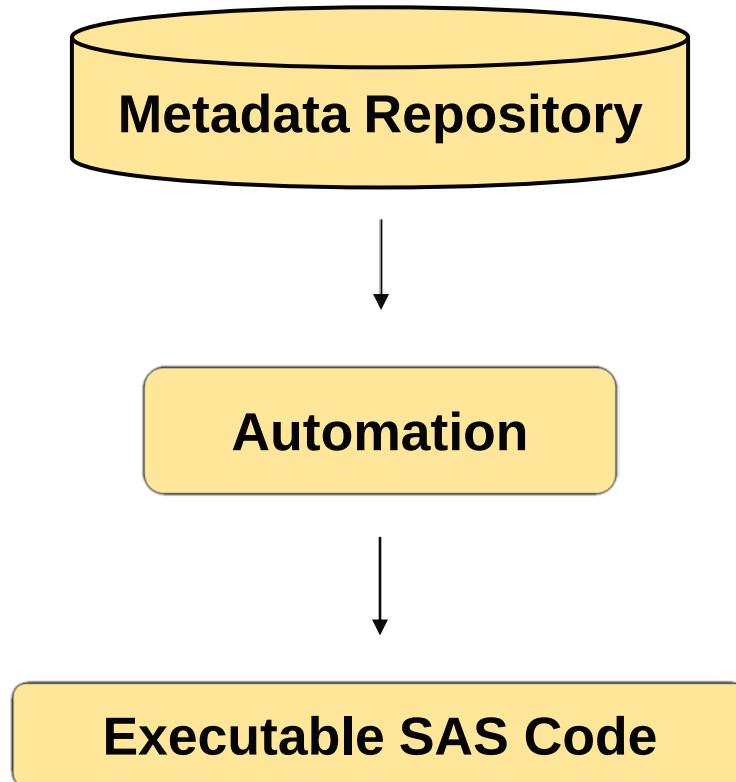
- ❑ The opinions in this presentation are those of the presenter and may not necessarily reflect the views of doLoopTech, PhUSE or CDISC.

- ❑ What is Metadata?
- ❑ High Level Automation
- ❑ Example of Typical SAS code
- ❑ Code Engineering
- ❑ Metadata for Wrapper Program
- ❑ Metadata for Macro Variable Definition
- ❑ Metadata for Joins
- ❑ Metadata for Target Variables
- ❑ Metadata for Target Structure
- ❑ Metadata for Codelist

What is Metadata?

- ❑ Metadata is most commonly defined as "data about data." In other words, metadata includes description of the content, context, structure and/or purpose of a database.





- ❑ Having a metadata driven system which creates the executable SAS code by reading the rules from a metadata repository.
- ❑ Also give enough flexibility for programmers to add their custom code into the process.

Example of a typical SAS code

*/*Create variables*/*

%global env type debug;

%let env = dev;
%let type = sdtm;
%let debug = Y;

*/*Setup libnames*/*

%setup;

*/*Read input data*/*

Data work.dm;
 set raw.dm;

...

..

Run;

*/*Join*/*

Proc SQL;

Create table work.dm_2 as

Select ...

From...

On ...

Quit;

*/*Derive target variables*/*

Data target.dm;
 set work.dm_2;

..

..

Run;

- ☐ Break your code into multiple generic modules.
- ☐ Try to convert each module into metadata.
- ☐ For example, a module can be:
 - Macro variable definition
 - Setup
 - Read Input Data
 - Joins
 - Derive Target Variables
- ☐ All the above metadata can be wrapped in a master metadata to create wrapper executable SAS program.

Metadata for Wrapper Program

Metadata for Wrapper Program

- ❑ This is the most important part of the metadata driven system.
- ❑ It describes the information needed to capture the macro calls and the parameters required to create the executable SAS code.
- ❑ It also describes the sequence in which the steps need to be written in the executable SAS code.
- ❑ Users can inject their custom step in between the existing flow.

Metadata for Wrapper Program

Metadata Variable	Description
OBJECT_PK	Primary Key of target object
STEP_NO	Step number
MACRO	SAS Macro name. This can be an external macro or one of the system code generator macro
PARAM	SAS Macro Parameter
PARAM_VALUE	SAS Macro Parameter Value

Metadata for Wrapper Program

OBJECT_PK	STEP_NO	MACRO	PARAM	PARAM_VALUE
1	1	setup.sas	raw_lib	/study1/data/raw
1	1	setup.sas	sdtm_lib	/study1/data/sdtm
1	2	readdata.sas	indsn	dm
1	2	readdata.sas	where_str	Age>20
1	3	<join_token>	outdsn	ex
1	4	<derive_token>	outdsn	sdtm.ae
1	5	<attrib_token>	outdsn	ae
1	6	post_processing.sas	indsn	sdtm.ae

/*Executable SAS Code*/

%setup(raw_lib=/study1/data/raw,
sdtm_lib=/study1/data/sdtm);

%readdata(indsn=dm, where_str=age>20);

Metadata for Wrapper Program

OBJECT_PK	STEP_NO	MACRO	PARAM	PARAM_VALUE
1	1	setup.sas	raw_lib	/study1/data/raw
1	1	setup.sas	sdtm_lib	/study1/data/sdtm
1	2	readdata.sas	indsn	Standard Validated Macros
1	2	readdata.sas	where_str	
1	3	<join_token>	outdsn	Machine Readable Code Generator
1	4	<derive_token>	outdsn	
1	5	<attrib_token>	outdsn	User Defined Custom Macros
1	6	post_processing.sas	indsn	

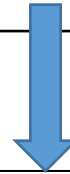
Metadata for Macro Variable Definition

Metadata for Macro Variable Definition

Metadata Variable	Description
OBJECT_PK	Primary Key of target object
NAME	Name of the macro variable
VALUE	Value of the macro variable
GLOBAL	Is this is a global variable (Y/N)

Metadata for Macro Variable Definition

OBJECT_ PK	NAME	VALUE	GLOBAL
1	env	dev	Y
1	type	sdtm	Y
1	debug	Y	Y



/*Executable SAS Code*/

%global env type debug;

%let env = dev;

%let type = sdtm;

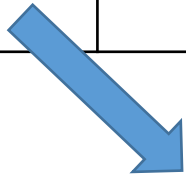
%let debug = Y;

Metadata for Joins

Metadata Variable	Description
OBJECT_PK	Primary Key of target object
OUTDSN	Output dataset name
LEFT	Left dataset name
LEFT_VARS	Left variable names for select statement
JOIN	Join type: INNER, LEFT, RIGHT
RIGHT	Right dataset name
RIGHT_VARS	Right variable names for select statement
JOIN_BY	Join by condition
WHERE	Where condition

Metadata for Joins

OBJECT_PK	OUTDSN	LEFT	LEFT_VARS	JOIN	RIGHT	RIGHT_VARS	JOIN_BY	WHERE
1	ex	raw.dm	dm.var1, dm.var2, dm.var3	LEFT	raw.ec	ec.var4, ec.var5	dm.subjid = ec.subjid	



*/*Executable SAS Code*/*

```
PROC SQL;  
CREATE TABLE work.ex as  
  SELECT dm.var1, dm.var2, dm.var3, ec.var4, ec.var5  
  FROM raw.dm LEFT JOIN raw.ec  
  ON dm.subjid = ec.subjid;  
QUIT;
```

Metadata for Target Variables

Metadata for Target Variables

Metadata Variable	Description
OBJECT_PK	Primary Key of target object
SOURCE	Source dataset name
SOURCE_VAR	Source variable name
RULE	Rule type
DERIVATION	Target variable derivation
TARGET	Target dataset name
TARGET_VAR	Target variable name
CODELIST_ID	Codelist ID as foreign key
IS_TEMP	Y/N flag to determine if this is a temporary derivation



Metadata for Target Variables

OBJECT_P K	SOURCE	SOURC E_VAR	RULE	DERIVATION	TARGET	TARGET_VAR	CODELIST_I D	IS_TEMP
1	raw.ae	studyid, subjid	derivation	trim(cat(strip(st udyid),"- ",strip(subjid)))	sdtm.ae	usubjid		N
1	raw.ae	ae	equal		sdtm.ae	aeterm		N
1	raw.ae	freq	derivation		sdtm.ae	aepatt	aepatt001	N

/*Executable SAS Code*/

data sdtm.ae;
set raw.ae;

usubjid = trim(cat(strip(studyid),"-",strip(subjid)));

aeterm = ae;

if freq = "" then aepatt = "";
else if freq = "1" then aepatt = "CONTINUOUS"
else if freq = "2" then aepatt = "INTERMITTENT";
else if freq = "U" then aepatt = "";

run;

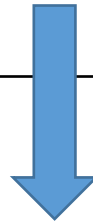
Metadata for Codelist

Metadata for Codelist

Metadata Variable	Description
CODELIST_ID	Primary Key of codelist
SOURCE_VAL	Source value
TARGET_VAL	Target value
TYPE	C2C, N2N, C2N, N2C

Metadata for Codelist

CODELIST_ID	SOURCE_VAL	TARGET_VAL	TYPE
aepatt001			C2C
aepatt001	1	CONTINUOUS	C2C
aepatt001	2	INTERMITTENT	C2C
aepatt001	U		C2C



*/*Executable SAS Code*/*

```
if freq = "" then aepatt = "";  
else if freq = "1" then aepatt = "CONTINUOUS"  
else if freq = "2" then aepatt = "INTERMITTENT";  
else if freq = "U" then aepatt = "";
```

Metadata for Target Structure

Metadata for Target Structure

Metadata Variable	Description
OBJECT_PK	Primary Key of target object
CDISC_VERSION	CDISC Version
DS_NAME	Dataset Name
VAR_ORDER	Variable Order
VAR_NAME	Variable Name
VAR_LABEL	Variable Label
VAR_TYPE	Variable Type
VAR_LENGTH	Variable Length (optional)
SORT_KEY	Order of sorting

Metadata for Target Structure

OBJECT_PK	CDISC_VERSION	DS_NAME	VAR_ORDER	VAR_NAME	VAR_LABEL	VAR_TYPE	VAR_LENGTH	SORT_KEY
1	SDTM 3.2.2	AE	1	STUDYID	Study Identifier	CHAR	200	1
1	SDTM 3.2.2	AE	2	DOMAIN	Domain Abbreviation	CHAR	200	
1	SDTM 3.2.2	AE	3	USUBJID	Unique Subject Identifier	CHAR	200	2
1	SDTM 3.2.2	AE	4	AESEQ	Sequence Number	NUM	8	
1	SDTM 3.2.2	AE	5	AEGRPID	Group ID	CHAR	200	

Metadata for Target Structure

/*Executable SAS Code*/

ATTRIB

studyid	label='Study Identifier'	length=\$200
domain	label='Domain Abbreviation'	length=\$200
usubjid	label='Unique Subject Identifier'	length=\$200
aeseq	label='Sequence Number'	length=8
aegrpid	label = 'Group ID'	length=\$200;

- ☐ By having a strong metadata driven system one can automate up to 80% of programming tasks.
- ☐ Always keep in mind the flexibility of the system. Programmers should be confident of the automated executable SAS code.
- ☐ Allow injecting of custom code in the metadata and not directly in the code.

Reach out to us @

Mousum Dutta

Director of Analytics

doLoop Technologies India Pvt. Ltd.

Email: mousum.dutta@dolooptech.com

www.dolooptech.com

Dr. Sangram Parbhane

Associate Clinical SAS Programmer

doLoop Technologies India Pvt. Ltd.

Email: sangram.parbhane@dolooptech.com

www.dolooptech.com



doLoop Technologies

Business Analytics. Delivered.

Thank You !