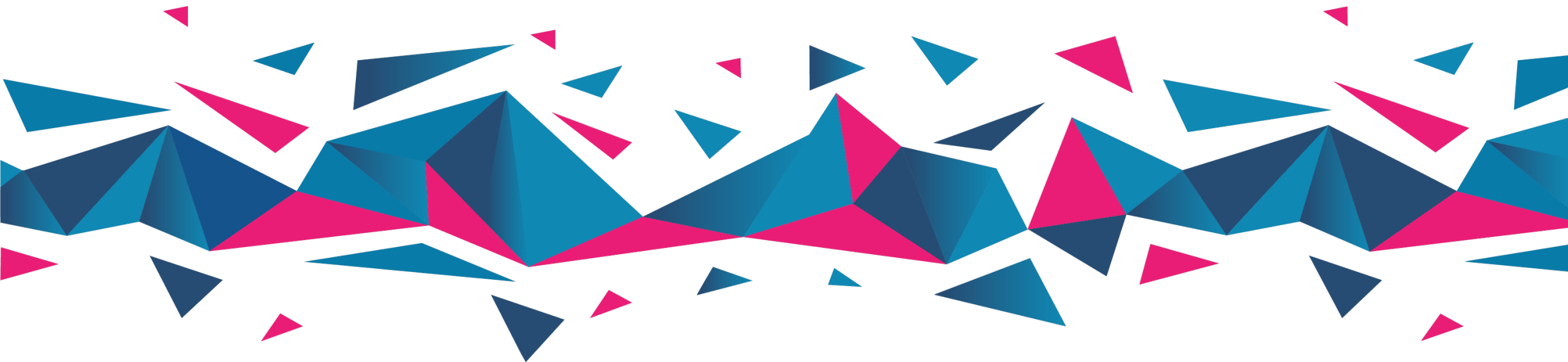


Advanced Excel Specification for Automatic SDTM Generation

Ilya Mishchenko & Aleksandra Gorbunova

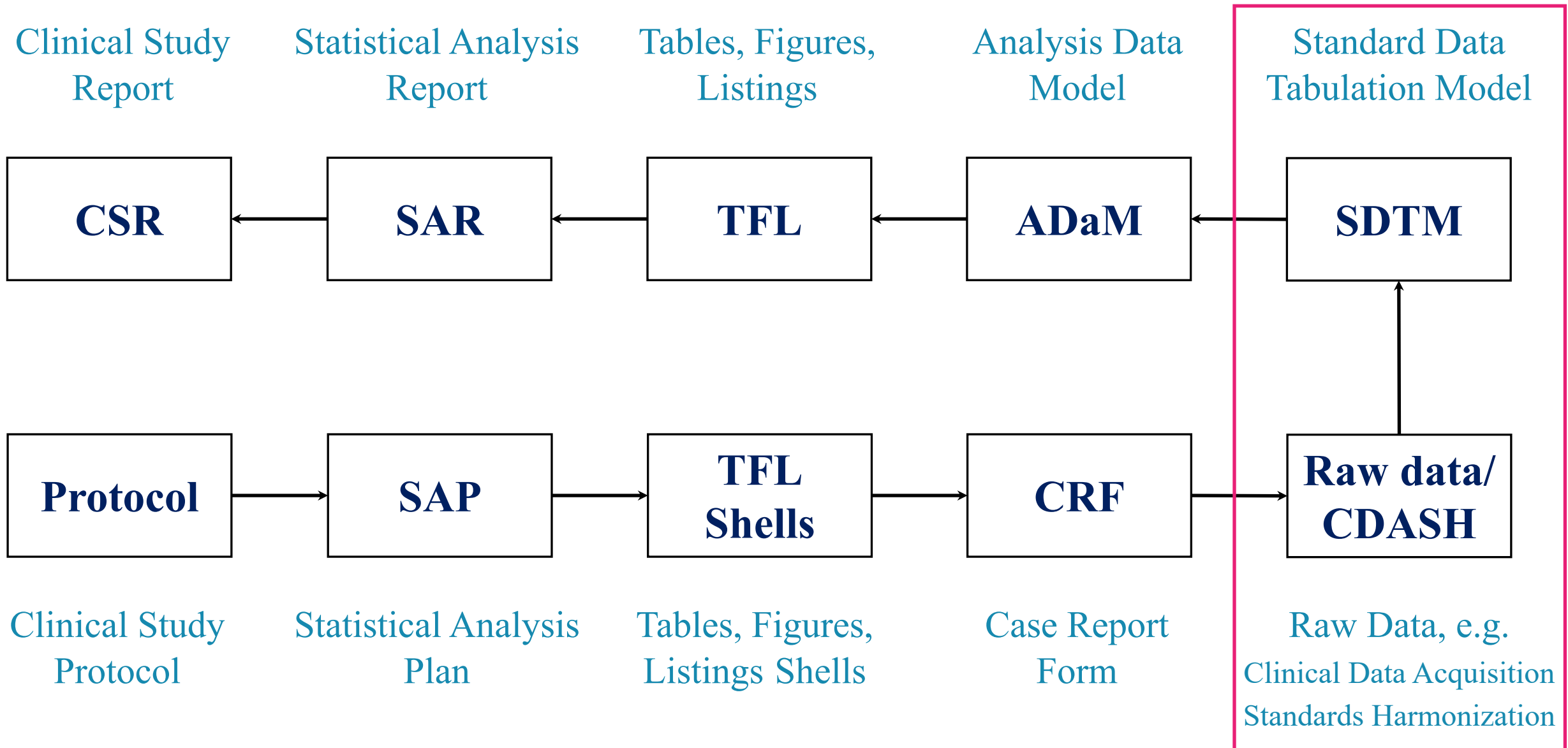
Co-authors: Evgeny Dmitriev & Timur Zagidullin



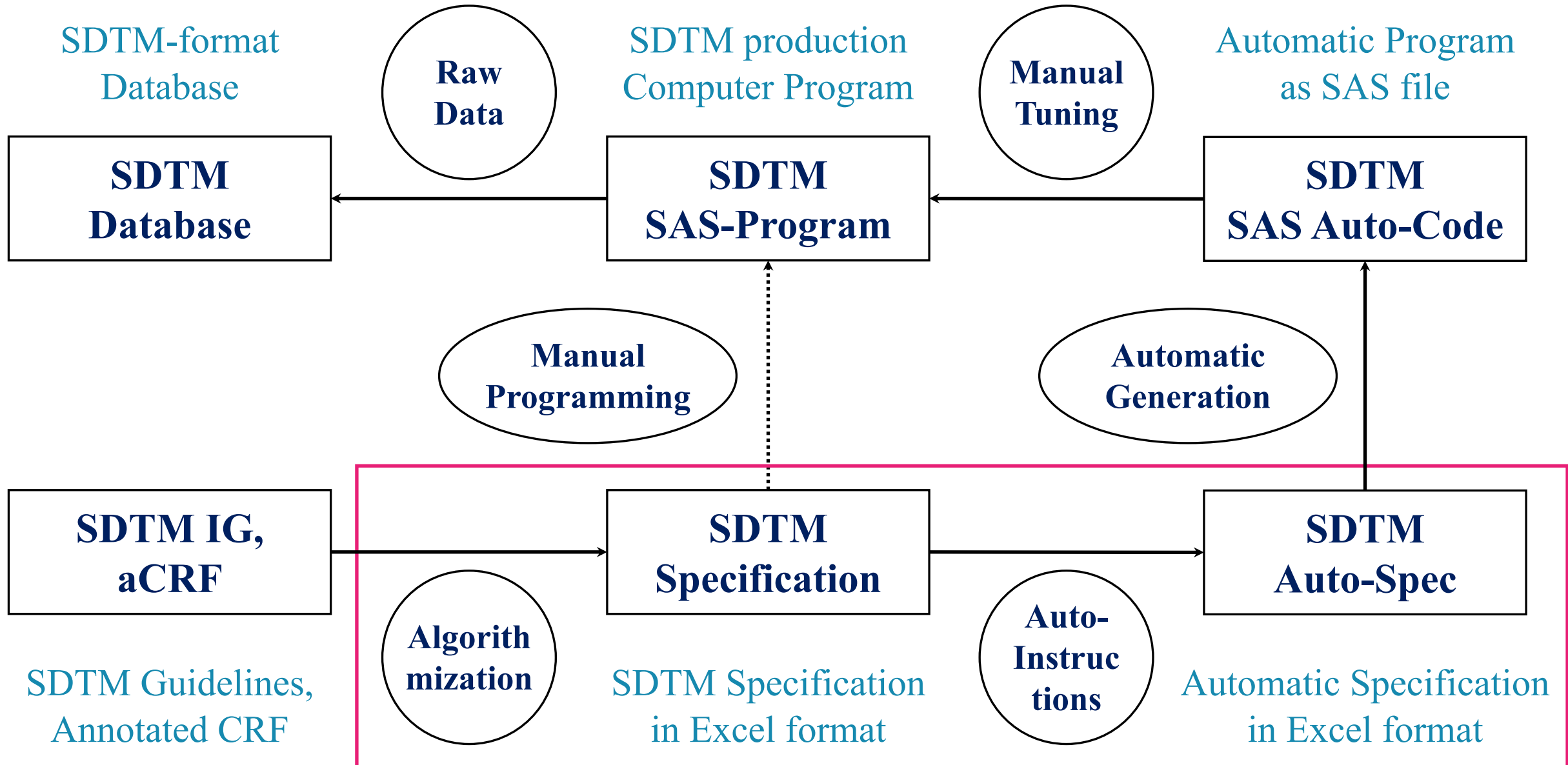
Agenda

- **Information-flow pipeline in the project life cycle**
- **SDTM-generating “junction” via standard Excel specification**
 - List of required domains
 - List of variables organized by domains
 - Set of predefined data-independent domains
- **Extension of the standard specification with SAS-instructions**
 - Local code-definition for each output variable
 - Data-step number for each output variable creation
 - Code surroundings (upper and lower borders) for each data-step
- **Three types of variable deriving instructions**
 - Straightforward derivations (inheriting, renaming and direct calculations)
 - SAS-arrays of variables and standard *do*-loops calculations
 - User-define macro-arrays of datasets and *%do*-loops calculations
- **Further development and conclusions**

Information-flow pipeline



SDTM generating



SDTM generating

List of domains

Standard instructions

Seq. For Order	Dataset	Description	Seq. For Program ming
1	AE	Adverse Events	5
2	CE	Clinical Events	16
3	CM	Concomitant Medications	27
4	DD	Death Details	8
5	DM	Demographics	1
6	DS	Disposition	3
7	DV	Protocol Deviations	10
8	EG	ECG Test Results	11
9	EX	Exposure	24

Extensions for auto-coding

Datasets sheet

SDTM generating

List of variables

Seq. For	Domain	Variable Name	Variable Label	Type	Controlled Terms	Role	Core	CDISC Notes	Origin	Source variable	Mapping type	Comments
1	AE	STUDYID	Study Identifier	Char		Identifier	Req	Unique identifier for a study.	Protocol		DIRECT	STUDYID = "Project"
2	AE	DOMAIN	Domain Abbreviation	Char	(DOMAIN)	Identifier	Req	Two-character abbreviation for the domain.	Assigned			DOMAIN = "AE"
3	AE	USUBJID	Unique Subject Identifier	Char		Identifier	Req	Identifier used to uniquely identify a subject across all studies for all applications or submissions involving the product.	DM	USUBJID	DIRECT	
4	AE	AESEQ	Sequence Number	Num		Identifier	Req	Sequence Number given to ensure uniqueness of subject records within a domain. May be any valid number.	Derived		DERIVATION	Sort records by USUBJID, AEDECOD, AESTDTC and assign AESEQ = 1 for the first record
5	AE	AETERM	Reported Term for the Adverse Event	Char		Topic	Req	Verbatim name of the event.	in_ae	AETERM	DIRECT	
6	AE	AELLTCD	Lowest Level Term Code	Num	MedDRA	Variable Qualifier	Exp	Dictionary-derived code for the Lowest Level Term.	in_ae	LLT_CODE	DIRECT	
7	AE	AELLT	Lowest Level Term	Char	MedDRA	Variable Qualifier	Exp	Dictionary-derived text description of the Lowest Level Term.	in_ae	LLT_NAME	DIRECT	

Variables sheet

SDTM generating

Set of data-independent domains

STUDY ID	RDOM AIN	USU BJID	IDVAR	IDVAR VAL	REL TYPE	REL ID
Project	PR		PRREFID		ONE	1
Project	TU		TUREFID		ONE	1
Project	PR		PRSPID		ONE	2
Project	RS		RSSPID		ONE	2

RELREC

STUDY ID	DOM AIN	IETEST CD	IETEST	IECAT	TIVERS
Project	TI	INCL01	Inclusion 1	INCLUSION	Amendment 1
Project	TI	INCL02	Inclusion 2	INCLUSION	Amendment 1
Project	TI	EXCL01	Exclusion 1	EXCLUSION	Amendment 1

TI

STUDY ID	DOM AIN	VISIT NUM	VISIT	ARM CD	ARM	TVSTR
Project	TV	1	Screening			Start of Screening Period
Project	TV	2	Product Manufacture			Product Manufacture
Project	TV	3	Pre-treatment Visit			10 days Prior to Initial Dose

TV

STUDY ID	DOM AIN	ARMCD	ARM	TAET ORD	ETCD	ELEMENT	TABR ANCH	TATR ANS	EPOCH
Project	TA	PRODCD	PRODUCT	1	SCRN	Screening			SCREENING
Project	TA	PRODCD	PRODUCT	2	PROD	Product every 2 weeks			TREATMENT
Project	TA	PRODCD	PRODUCT	3	FU	Follow-up			FOLLOW-UP

TA

STUDY ID	DOM AIN	ETCD	ELEMENT	TESTRL	TEENRL	TED UR
Project	TE	SCRN	Screening	Informed Consent	Screening assessments are complete	
Project	TE	PROD	Product every 2 weeks	First dose of treatment every 2 weeks	8 weeks after start of Element	P8W
Project	TE	FU	Follow-up	Follow Up Visits	1 year after start of Element	P1Y

TE

STUDY ID	DOM AIN	TSS EQ	TSGR PID	TSPARM CD	TSPARM	TSVAL
Project	TS	1		AGEMIN	Planned Minimum Age of Subjects	P18Y
Project	TS	1		LENGTH	Planned Trial Length	P64W
Project	TS	1		ACTSUB	Actual Number of Subjects	XXX

TS

Extension with SAS instructions

Local code-definition

Seq. For	Domain	Variable Name	Variable Label	Type	Code Substitutions
30	AE	AESMIE	Other Medically Important Serious Event	Char	AESMIE = ifc (AESER2Y = "Other Medically Important Event", "Y", "");
31	AE	AESTDTC	Start Date/Time of Adverse Event	Char	AESTDTC = AESTDAT_DTS;
32	AE	AEENDTC	End Date/Time of Adverse Event	Char	AEENDTC = AEENDAT_DTS;
33	AE	AESTDY	Study Day of Start of Adverse Event	Num	AESTDY = %DateInt(RFICDTN, datepart (AESTDAT));
34	AE	AEENDY	Study Day of End of Adverse Event	Num	AEENDY = %DateInt(RFICDTN, datepart (AEENDAT));

Variables sheet

Macro Functions
/* Date Interval */
%macro DateInt (StartDate, endDate); /* Start Date, End Date */
&endDate. - &StartDate. + (&endDate. >= &StartDate.)
%mend;

Macro sheet

Extension with SAS instructions

Data-step number & Code surroundings (1/2)

Seq. For	Domain	Variable Name	Variable Label	Type	Data Step	Code Surroundings
						<pre> data Libr.&&SDTMName&d..1; merge Libr.&&SDTMName&d..0 Libr.in_ae; keep %LoopSDTM_InVar_n (&d, %nrstr(&&SDTMVarName&d._&n.)) AESTDAT AEENDAT /* For Further Calculations */ VMEDDRA AEDLT AEIM; /* For SUPPAE Domain */ /* First One-String File merge, no by-statement */ </pre>
5	AE	AETERM	Reported Term for the Adverse Event	Char	1	
32	AE	AEENDTC	End Date/Time of Adverse Event	Char	1	<pre> proc sort; by &InIDVar. AEDECOD AESTDTC; run; </pre>

Variables sheet

Extension with SAS instructions

Data-step number & Code surroundings (2/2)

Macro Functions

/* Macro-Loop over SDTM Variables */

```
%macro LoopSDTM_Var_n(d, MaskStat);    /* Domain Number, Statement Masked by %nrstr() */
%do n = 1 %to &&SDTMVarNum&d.;          /* From Specification Analysis */
%unquote (&MaskStat.)
%end;
%mend;
```

/* Macro-Loop over SDTM Variables with Input Datasets ID-Variable */

```
%macro LoopSDTM_InVar_n(d, MaskStat);    /* Domain Number, Statement Masked by %nrstr() */
&InIDVar.                                  /* Input-ID Variable */
%LoopSDTM_Var_n (&d., &MaskStat.)
%mend;
```

Macro sheet

&d.	↔	Seq. For Order column,	&&SDTMName&d.	↔	Dataset column	in Datasets sheet
&n.	↔	Seq. For Order column,	&&SDTMVarName&d._&n.	↔	Variable Name column	in Variables sheet
			&&SDTMVarNum&d.	=	Number of variables	for current domain
			%macro LoopSDTM_InVar_n (d, MaskStat)	–	Macro-Loop over variables	for current domain

Variable deriving instructions

Straightforward derivations

Seq. For	Domain	Variable Name	Variable Label	Type	Code Substitutions
5	AE	AETERM	Reported Term for the Adverse Event	Char	<code>/* Automatically Filled-In */</code>
7	AE	AELLT	Lowest Level Term	Char	<code>AELLT = LLT_NAME;</code>
19	AE	AESER	Serious Event	Char	<code>AESER = put (AESER, \$CTFmtYN.);</code>

Inheriting
Renaming
Direct calculations

Variables sheet

Project Formats
<pre>proc format; value \$CTFmtYN /* Yes/No */ "Yes" = "Y" "No" = "N" ;</pre>

Formats sheet

Variable deriving instructions

SAS-arrays and *do*-loops (1/2)

Seq. For	Domain	Variable Name	Variable Label	Type	Data Step	Code Surroundings
						<pre> data Libr.&&SDTMName&d..1; merge Libr.&&SDTMName&d..0 Libr.AE_; keep %LoopSDTM_InVar_n (&d., %nrstr(&&SDTMVarName&d._&n..)); /* First One-String File merge, no by-statement */ </pre>
4	SUPP AE	IDVAR	Identifying Variable	Char	1	<pre> %let NVal = 3; /* Number of Assigned Variables Values */ </pre>

Seq. For	Domain	Variable Name	Variable Label	Type	Code Substitutions
6	SUPP AE	QNAM	Qualifier Variable Name	Char	<pre> array NameVal [&NVal] &CharVarFormat._temporary_ ("AEDICT" "AEDLT" "AEIMMU"); array LabelVal [&NVal] &CharVarFormat._temporary_ ("Dictionary Version" </pre>
7	SUPP AE	QLABEL	Qualifier Variable Label	Char	<pre> "Is this a suspected DLT?" "Is adverse event immune related?"); </pre>
8	SUPP AE	QVAL	Data Value	Char	<pre> array ValueVar [&NVal] &CharVarFormat. VMEDDRA AEDLT AEIM ; </pre>

Variable deriving instructions

SAS-arrays and *do*-loops (2/2)

Seq. For	Domain	Variable Name	Variable Label	Type	Code Substitutions	Data Step	Code Surroundings
					<pre> do I = 1 to &NVal; QNAM = NameVal [I]; QLABEL = LabelVal [I]; QVAL = ValueVar [I]; QEVAL = ifc (QNAM in ("AEDLT","AEIMMU"), "INVESTIGATOR", ""); output; end;</pre>		
10	AE	QEVAL	Evaluator	Char		1	run;

Variables sheet

Project Constants
<pre> /* Numerical and Character Variable Lengths and Formats */ %let NumVarLength = 8; %let CharVarLength = 200; %let NumVarFormat = Best.; %let CharVarFormat = \$&CharVarLength..;</pre>

Constants sheet

Variable deriving instructions

User-define macro-arrays and %do-loops (1/5)

AnNum	AnCode	AnName	AnDS
<i>Analysis Number</i>	<i>Analysis Code</i>	<i>Analysis Name</i>	<i>Analysis Dataset</i>
1	HEM	HEMATOLOGY	in_hem
2	CHEM	CHEMISTRY	in_chem
3	CO	COAGULATION	in_co

Laboratory Analysis sheet

AnNum	AnCode	ParNum	ParCode	ParName
<i>Analysis Number</i>	<i>Analysis Code</i>	<i>Parameter Number</i>	<i>Parameter Code</i>	<i>Parameter Name</i>
HEMATOLOGY				
1	HEM	1	HCT	HEMATOCRIT
1	HEM	2	HGB	HEMOGLOBIN
1	HEM	3	MCV	ERY. MEAN CORPUSCULAR VOLUME
CHEMISTRY				
2	CHEM	12	ALB	ALBUMIN
2	CHEM	13	ALP	ALKALINE PHOSPHATASE
2	CHEM	14	ALT	ALANINE AMINOTRANSFERASE

Laboratory Parameters sheet

Variable deriving instructions

User-define macro-arrays and %do-loops (2/5)

Domain	Data Step	Code Surroundings
		<pre> /* Macro-Loop over All Laboratory Analysis Datasets */ %MacroLoopAn_An (%nrstr (data Libr.&&SDTMName&d._&An; merge Libr.&&SDTMName&d..0 Libr.&&AnDS&An.; keep %LoopSDTM_InVar_n (&d., %nrstr(&&SDTMVarName&d._&n..)) LBSign; /* For SUPPLB Domain */ /* First One-String File merge, no by-statement */ /* Macro-Loop over All Laboratory Parameters within Current Analysis */ LB 1 %MacroLoopPar_k (&An, %nrstr (</pre>

Seq. For	Domain	Variable Name	Variable Label	Type	Code Substitutions
5	LB	LBTESTCD	Lab Test or Examination Short Name	Char	LBTESTCD = "&&ParCode&An. &k.";
6	LB	LBTEST	Lab Test or Examination Name	Char	LBTEST = "&&ParName&An. &k.";
7	LB	LBCAT	Category for Lab Test	Char	LBCAT = "&&AnName&An.";

**Variables
sheet**

Variable deriving instructions

User-define macro-arrays and %do-loops (3/5)

Domain	Data Step	Code Surroundings
		<pre> output;)) run;)) /* Combining Prepared Laboratory Domains for All Analysis Together */ data Libr.&&SDTMName&d..1; set %MacroLoopAn_An (%nrstr (Libr.&&SDTMName&d.._&An.)); proc sort; by &InIDVar. LBCAT LBTESTCD LBDTC; run; </pre>
LB	1	

Variables sheet

Variable deriving instructions

User-define macro-arrays and %do-loops (4/5)

Macro Functions

```
/** LB Domain **/
```

```
/* Macro-Loop over All Analysis */
```

```
%macro MacroLoopAn_An (MaskStat); /* Statement Masked by %nrstr() */  
%do An = 1 %to &NAn.; /* Forming Through Laboratory Analysis Specification */  
  %unquote (&MaskStat.)  
%end;  
%mend;
```

```
/* Macro-Loop over Parameters for Specified Analysis */
```

```
%macro MacroLoopPar_k (An, MaskStat); /* Analysis, Statement Masked by %nrstr() */  
%do k = 1 %to &&NPar&An.; /* Forming Through Laboratory Parameters Specification */  
  %unquote (&MaskStat.)  
%end;  
%mend;
```

Macro sheet

Variable deriving instructions

User-define macro-arrays and %do-loops (5/5)

Laboratory Analysis sheet:

&An. ↔ **Analysis Number** column

&&AnDS&An. ↔ **Analysis Dataset** column

&&AnName&An. ↔ **Analysis Name** column

&NAn. = Number of Laboratory Analysis Types

%macro MacroLoopAn_An (MaskStat) –

Macro-Loop over Laboratory Analysis Types

Laboratory Parameters sheet:

&k. ↔ **Parameter Number** column

&&ParCode&An_&k. ↔ **Parameter Code** column

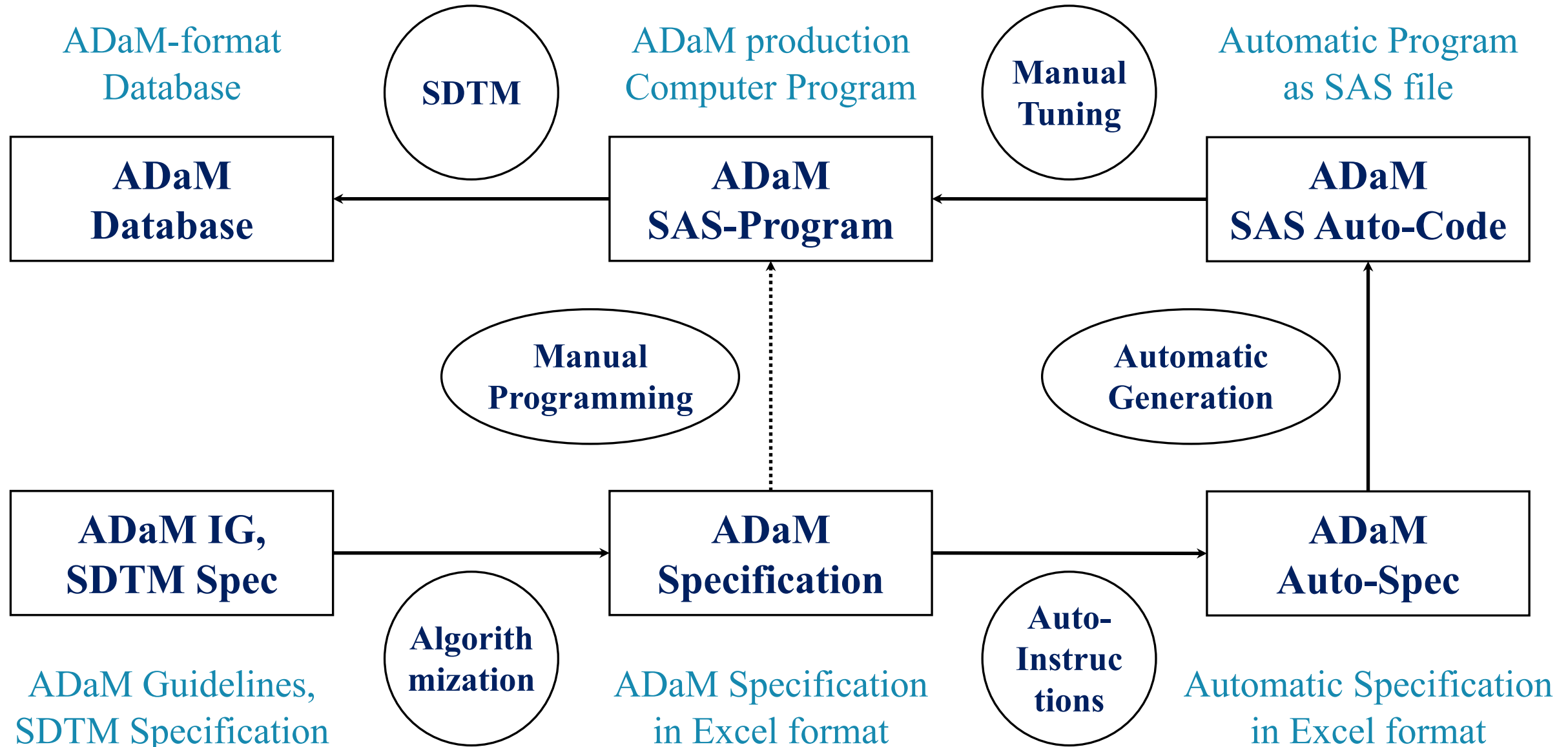
&&ParName&An_&k. ↔ **Parameter Name** column

&&NPar&An. = Number of Laboratory Parameters for Specified Analysis Type

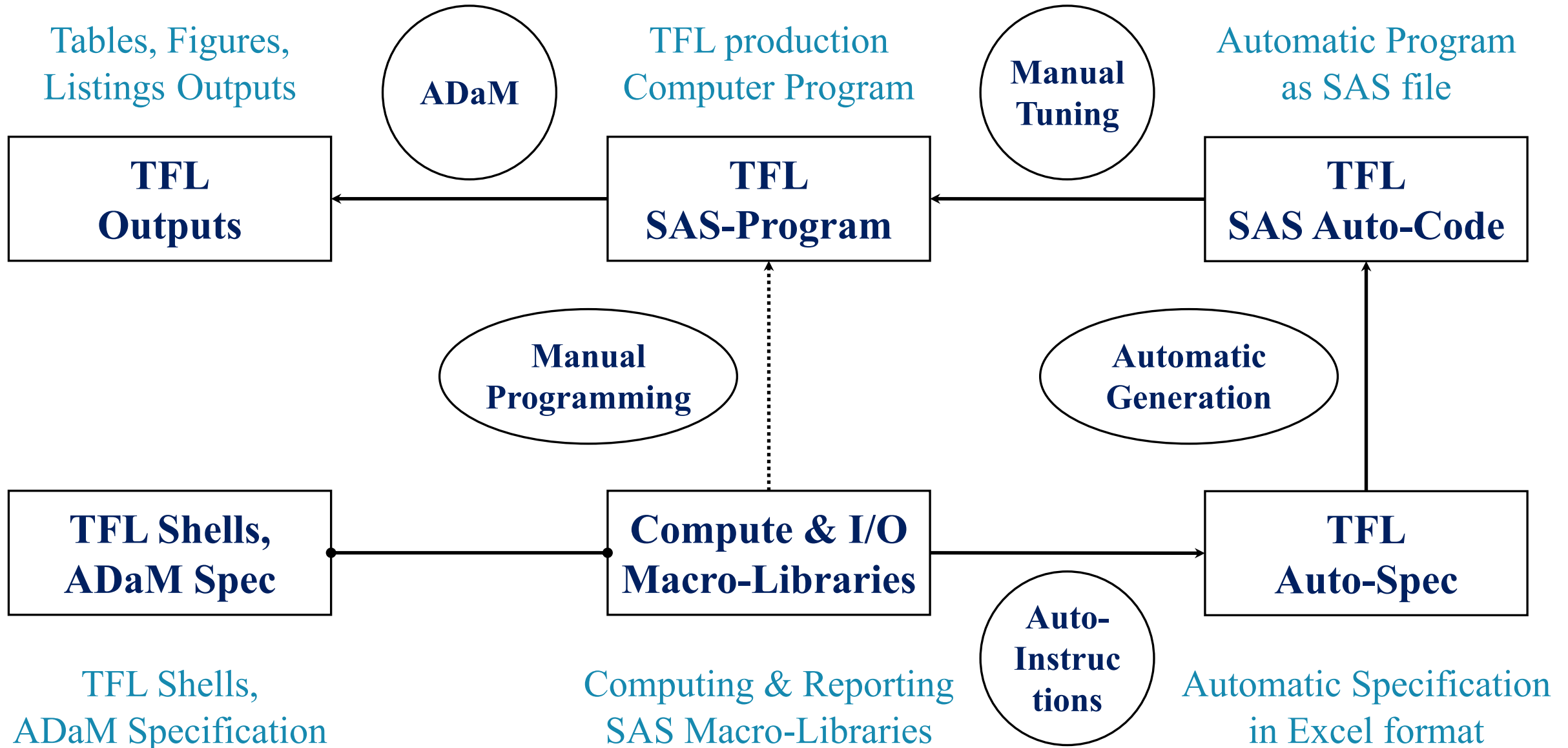
%macro MacroLoopPar_k (An, MaskStat) –

Macro-Loop over Laboratory Parameters for Specified Analysis Type

Further development: ADaM generating



Further development: TFL generating



Conclusions

- **General framework of information-flow pipeline in the project life cycle is outlined**
- **Novel approach to automatic SDTM generation by extension of standard Excel specification with SAS-code instructions is proposed**
- **Three types of variables deriving instructions which are generally sufficient for SDTM construction are determined**
- **Application of the proposed approach to ADaM and TFL automated production is considered**

Contacts

Ilya Mishchenko

ilya.mishchenko@atlantclinical.com

Aleksandra Gorbunova

aleksandra.gorbunova@atlantclinical.com

Evgeny Dmitriev

evgeny.dmitriev@atlant-clinical.com

Timur Zagidullin

timur.zagidullin@atlantclinical.com