Paper SI09

"It's ARM, Jim, but not as we know it" - our journey to automating output delivery through metadata

Edwin van Stein, GSK, Netherlands
Warwick Benger, GSK, United Kingdom

ABSTRACT

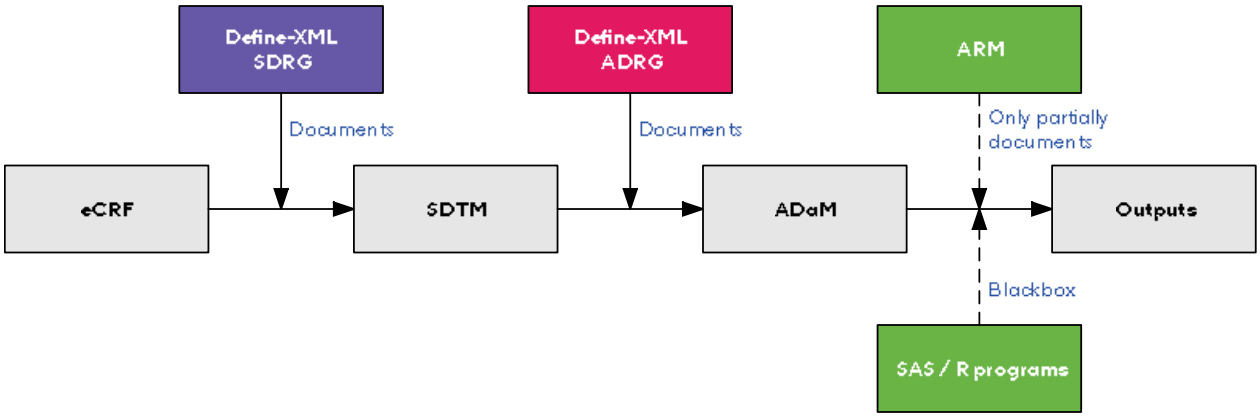
To improve traceability, transparency and consistency GSK is implementing automation of analyses and output creation through standardized metadata. Tool-agnostic Analysis Results Metadata (not to be confused with CDISC ARM!) is used to produce Analysis Results Data. Following the WORM principle (Write Once Read Many) tool-agnostic Analysis Output Metadata then uses those results to render them to different displays (tables, figures and listings, both post-text and in-text), other outputs (e.g., XML files for disclosure) and systems (in whatever format those systems require) without reproducing the numbers (and potentially getting them wrong!). This leads to a process where it's easier to validate and reuse results and to build tools in a multilingual setting. This paper focuses on the metadata, the structure of the Analysis Results Data and how it all fits together on our journey to standardizing our results.

DISCLAIMER

The content of this paper is very much a work in progress. By the time you read this, our internal version will look different. Feel free to reach out to the author (see end of this paper for contact details) with any comments or questions.

INTRODUCTION

Traceability and transparency of collected data and analysis data are ensured by using CDISC's SDTM and ADaM standards and are documented in Define-XML and the SDRG and ADRG. The analysis of primary endpoints can be traced back to ADaM using ARM in Define-XML, but any analysis not defined in ARM has no clear traceability through submission deliverables. SAS or R programs used to create outputs are often part of the submission package and can provide some traceability, but if they use a lot of macros or packages then they become something of a blackbox. In addition, ARM is very often created retrospectively and is not necessarily meant to be used to drive automation.

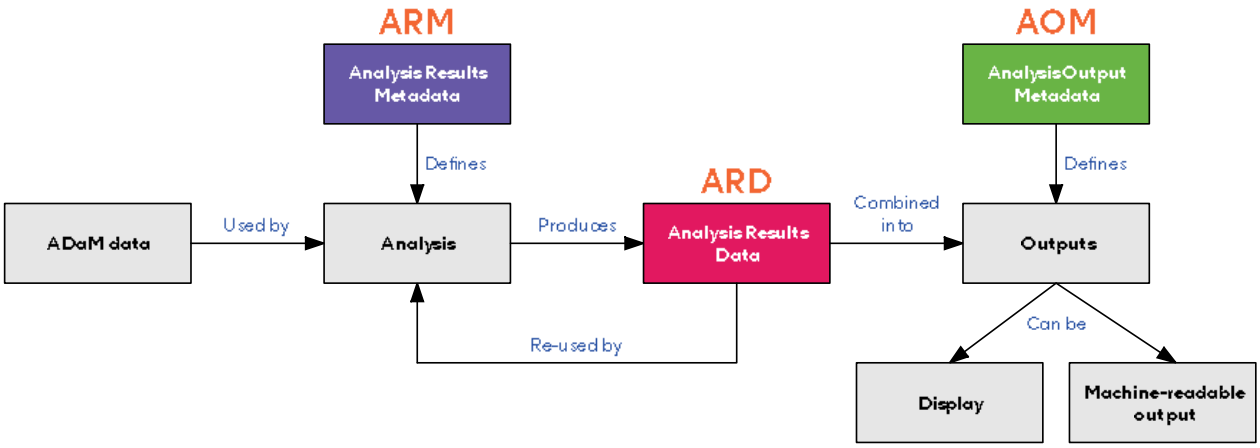


Within the broader End-to-End Standards project at GSK, we're looking into defining metadata that describes the path from analysis data to outputs for increased traceability, transparency, automation, and consistency.

WORM

The central principle behind our implementation is Write Once, Read Many (WORM). Any analysis should be defined, executed, and validated only once and then the numbers should be re-used across outputs and analyses. This also applies to simple analyses like a frequency count analysis on the number of subjects in the safety analysis set by treatment. This analysis should only be performed once and then be re-used as denominator and in column headers across many outputs.

OUR VISION



Analysis Results Metadata (ARM, not to be confused with CDISC ARM!) defines a single analysis, for instance the calculation of analysis set counts by treatment. Standard tools consume the ARM and ADaM data to produce Analysis Results Data (ARD). As mentioned before the ARD can be re-used by another analysis, for instance by using the analysis set counts by treatment as denominator in another analysis.

Analysis Output Metadata (AOM) defines an output, for instance a demography table. It defines which sets of ARD should be included, how to combine them and how to format them. Standard tools consume the AOM and ARD to then produce the outputs. Creating a slightly different in-text version of an already defined end-of-text table is then just a matter of defining the AOM to produce that table without re-doing the entire analysis. The same Analysis Results Data can also be used to feed into other systems, for instance as an XML file for disclosure purposes or as JSON file for CSR automation.

DEFINITION OF ANALYSIS

An analysis is defined as a single analysis on one variable using one method. Some examples of distinct analyses include:

- Frequency counts by treatment (e.g., big N)
- Frequency counts by treatment and sex
- Frequency counts by treatment, sex, and age group
- Summary statistics on age by treatment
- Summary statistics on age by treatment and sex

Table X: Summary of Demographic and Baseline Characteristics

	Treatment A (N=XXX)	Treatment B (N=XXX)	Total (N=XXX)
Sex			
n	xxx	xxx	xxx
F	xx (xx%)	xx (xx%)	xx (xx%)
M	xx (xx%)	xx (xx%)	xx (xx%)
Age (YEARS) at First Dose			
n	xxx	xxx	xxx
Mean	xx	xx	xx
SD	xx	xx	xx

Looking at the very basic demography table above four distinct analyses can be identified:

1. Frequency counts by treatment
2. Frequency counts by treatment for subjects with non-missing sex
3. Frequency counts by treatment and sex
4. Summary statistics on age by treatment

In this example the first analysis is only used for column headers, but without re-using this analysis we would be generating the numbers (big N) very often to be used as column headers or denominator in other analyses. With the WORM principle we have chosen to only generate and validate the numbers once and then re-use them. The second analysis is used as denominator for the third analysis.

ANALYSIS CONCEPTS

There is a lot of information that we want to capture in our metadata that we re-use across analyses and outputs. For instance, most analyses will use TRTAN and TRTA as by variables and the safety analysis set is always defined as SAFFL="Y". In addition, whenever we use TRTA we know that we want to sort by TRTAN. To re-use this information, we have defined some Analysis Concepts.

Analysis Variable Groups

Analysis Variable Groups are used to link variables, their units and their codelists. These are used by other concepts, ARM, and AOM. Examples of these are in the table below.

Variable GroupID	Variable GroupLabel	Variable Num	Variable Code	Variable Decode	Variable Unit	Codelist
TRTAN_TRTA	Actual Treatment	TRTAN		TRTA		TRT
TRTPN_TRTP	Planned Treatment	TRTPN		TRTP		TRT
ATOXGR_ATOXGRN	Toxicity Grade	ATOXGRN		ATOXGR		ATOXGR
AAGE1_AAGE1U	Age at First Dose	AAGE1			AAGE1U	
SEXN_SEX	Sex	SEXN	SEX			SEX
PARAMN_PARAM	Parameter	PARAMN	PARAMCD	PARAM		PARAM
AVISITN_AVISIT	Visit	AVISITN		AVISIT		AVISIT
ATPTN_ATPT	Timepoint	ATPTN		ATPT		ATPT

If we now use Actual Treatment as by variable in an analysis, then we know that TRTA and TRTAN are used. If we then use the same concept as column variables in a table, then we know that those columns will be sorted by TRTAN and labeled using the TRT codelist. Similarly, if we do an analysis on Age at First Dose then we know that the unit corresponding to the AAGE1 variable is AAGE1U.

Analysis Variable Matrix

An Analysis Variable Matrix is used to define combinations of variables or variable groups and whether a full matrix should be created when using these as by variables. These are used by ARM and AOM. Examples of these are in the table below.

Variable MatrixID	VariableMatrixLabel	Variables	FullMatrix
AVISIT_ATPT	Visit, Timepoint	AVISITN_AVISIT , ATPTN_ATPT	N
SEX_FULL	Sex (full matrix)	SEXN_SEX	Y
TRTA_FULL	Actual Treatment (full matrix)	TRTAN_TRTA	Y
SOC	System Organ Class	AESOC	N

Variable MatrixID	VariableMatrixLabel	Variables	FullMatrix
PT	Preferred Term	AEDECOD	N

If we now use AVISIT_ATPT as by variables in an analysis, then the previously defined variable groups AVISITN_AVISIT and ATPTN_ATPT are used. In addition, only the value combinations of these variables present in the data end up in the resulting ARD since full matrix was not specified. For SEX_FULL the full matrix is specified which means that the tools look up the codelist (defined in the variable group SEXN_SEX) and in the resulting ARD all possible values for sex are stored even if they do not appear in the analyzed data.

We don't want users to have to remember the ID of the different concepts. In the tools that need to be developed for maintaining this metadata we want users to be able to select concepts based on the labels. So, where AVISIT_TPT refers to AVISITN_AVISIT the user will see that the variable matrix called Visit, Timepoint uses the variable group Visit.

Analysis Range Checks

An Analysis Variable Matrix is used to define range checks (a.k.a. where clauses). These are used by ARM and AOM. Examples of these are in the table below.

RangeCheckID	RangeCheckLabel	RangeCheck
SAF	Safety Analysis Set	SAFFL EQ "Y"
TRTEM	Treatment Emergent	TRTEMFL EQ "Y"
SEX_MF	Sex (Male, Female)	SEX IN ("M", "F")

To ensure the range checks are CDISC compliant some rules are applicable:

- Only EQ, GT, GE, LT, LE, NE, IN, NOTIN are allowed
- Only the AND operator is allowed
- Only parenthesis for IN and NOTIN are allowed

If a range check needs anything more complex than that should be done by defining additional variables and flags in ADaM.

ANALYSIS RESULTS METADATA

ARM is split up into two parts, duplication ARM and main ARM.

Duplication ARM

For some analyses duplication of data is needed before running the analysis. Examples include duplicating all records to produce a total column or duplicating certain records to produce aggregate categories. We don't always want to do these duplications in ADaM. To duplicate all records with TRTAN

values 1 and 2, change TRTAN to 9 and the corresponding decode (defined in the variable group to be TRTA) to Total the duplication ARM would look something like below.

Duplication GroupID	Variables	Input Num	Input Code	Duplicated Num	Duplicated Code	Duplicated Decode
TRTA_TOT	TRTAN_TRTA	1		9		Total
TRTA_TOT	TRTAN_TRTA	2		9		Total

Another example of duplicating toxicity grades to an aggregate category can be seen below.

Duplication GroupID	Variables	Input Num	Input Code	Duplicated Num	Duplicated Code	Duplicated Decode
ATOXGR_345	ATOXGRN_ATOXGR	3		8		Grade 3+4+5
ATOXGR_345	ATOXGRN_ATOXGR	4		8		Grade 3+4+5
ATOXGR_345	ATOXGRN_ATOXGR	5		8		Grade 3+4+5

Duplications are applied sequentially. So, if both examples are applied to a data set with a record that has TRTAN=1 and ATOXGRN=3 then that record is quadrupled:

- TRTAN=1 and ATOXGRN=3 (the original record)
- TRTAN=9 and ATOXGRN=3 (duplicated on TRTAN)
- TRTAN=1 and ATOXGRN=8 (duplicated on ATOXGRN)
- TRTAN=9 and ATOXGRN=8 (duplicated on both TRTAN and ATOXGRN)

Main ARM

The main ARM contains one record per analysis and refers to Analysis Concepts and Duplication ARM. The ARM record for the frequency counts by treatment contains:

Variable	Value	Explanation
ARMID	AS_SAF	a unique ID to refer to the ARM record
DuplicationGroupID	TRTA_TOT	a link to duplication ARM defined previously
Domain	ADSL	the input data set
AnalysisSet	SAF	a link to the Safety Analysis Set concept define previously
Method	CountDistinctSubject	see below
ByVariables	TRTA_FULL	a link to the Actual Treatment (full matrix) concept defined previously

We want the metadata to be tool agnostic, so the method, CountDistinctSubject, comes from a pre-defined list of methods that can be used and the tools translate that into actionable code.

The ARM for the frequency counts by treatment for subjects with non-missing sex is very similar to the previous example:

Variable	Value
ARMID	DM_SEX_ANY
DuplicationGroupID	TRTA_TOT
Domain	ADSL
AnalysisSet	SAF
SubSet	SEX_MF
Method	CountDistinctSubject
ByVariables	TRTA_FULL

The only thing new is a reference to the Sex (Male, Female) range check to only select subjects with non-missing sex.

For the frequency counts by treatment and sex the ARM record is again very similar:

Variable	Value
ARMID	DM_SEX
DenomID	DM_SEX_ANY
DuplicationGroupID	TRTA_TOT
Domain	ADSL
AnalysisSet	SAF
SubSet	SEX_MF
Method	CountDistinctSubject
ByVariables	TRTA_FULL * SEX_FULL

The denominator is now defined by specifying the previously analysis as DenomID. In addition, the by variables are now both the Actual Treatment (full matrix) and the Sex (full matrix) concepts. This means that in the ARD we'll have records for all possible combinations of TRTAN, TRTA, SEXN and SEX based on their codelists regardless of whether those combinations exist in the input data.

Summary statistics are not that different from frequency counts in ARM, for the summary statistics on age by treatment the ARM is:

Variable	Value
ARMID	DM_AGE
DuplicationGroupID	TRTA_TOT
Domain	ADSL
AnalysisSet	SAF
Method	SummaryStat
AnalysisVariable	AAGE1_AAGE1U
ByVariables	TRTA_FULL

Obviously, the method is different. In this case the SummaryStat method produces a default list of relevant summary statistics. If any of these statistics don't have to be reported, then these can be filtered out in AOM later. The entire list will be produced so that if we want to replay the results to a different output with additional statistics then we don't have to re-run and re-validate the analysis.

For the SummaryStat method we also need to define the analysis variable. In this case it's the Age at First Dose concept. By using this concept we can specify at AOM later that the unit should be printed without specifying which variable holds the unit again as that it already done in the concept.

Combining the above the complete ARM for the basic demography table is (concepts highlighted red):

ARMID	DenomID	Duplication GroupID	Domain	Analysis Set	SubSet	Method	AnalysisVariable	ByVariables
AS_SAF		TRTA_TOT	ADSL	SAF		CountDistinctSubject		TRTA_FULL
DM_SEX_ANY		TRTA_TOT	ADSL	SAF	SEX_MF	CountDistinctSubject		TRTA_FULL
DM_SEX	DM_SEX_ANY	TRTA_TOT	ADSL	SAF	SEX_MF	CountDistinctSubject		TRTA_FULL * SEX_FULL
DM_AGE		TRTA_TOT	ADSL	SAF		SummaryStat	AAGE1_AAGE1U	TRTA_FULL

ANALYSIS RESULTS DATA

For ARD we have decided to store the results in a normalized structure so that regardless of the method or by variables the structure is always the same. It is a machine-readable structure and not necessarily a human-readable structure as the data is meant to be consumed by tools to produce human-readable outputs. Important metadata from ARM and concepts are not repeated in ARD as ARD is not meant to be consumed without the corresponding ARM. The ARD for the frequency counts by treatment are shown in the table below.

ARMID	QualifierID	Role	Variable	ValueNum	ValueCode	ValueDecode
AS_SAF	1.01	ByVariable	TRTAN:TRTA	1		Treatment A
AS_SAF	1.01	Statistic		49	N	
AS_SAF	2.01	ByVariable	TRTAN:TRTA	2		Treatment B
AS_SAF	2.01	Statistic		51	N	
AS_SAF	3.01	ByVariable	TRTAN:TRTA	9		Total
AS_SAF	3.01	Statistic		100	N	

The variables present in ARD are self-explanatory except for QualifierID. The integer to the left of the decimal dot groups related rows. In this case we know that the by variables from the first row and the statistic from the second row belong to each other. The decimals are used to group different statistics relating to the same result. The next example for the frequency counts by treatment and sex illustrates this:

ARMID	QualifierID	Role	Variable	ValueNum	ValueCode	ValueDecode
DM_SEX	1.01	ByVariable	TRTAN:TRTA	1		Treatment A
DM_SEX	1.01	ByVariable	SEXN:SEX	1	M	
DM_SEX	1.01	Statistic		25	N	
DM_SEX	1.02	ByVariable	TRTAN:TRTA	1		Treatment A
DM_SEX	1.02	ByVariable	SEXN:SEX	1	M	
DM_SEX	1.02	Statistic		49	Denom	
DM_SEX	1.03	ByVariable	TRTAN:TRTA	1		Treatment A
DM_SEX	1.03	ByVariable	SEXN:SEX	1	M	
DM_SEX	1.03	Statistic		51.0204081633	Percent	

In this case all nine records are related statistics for the same result. The first three rows are for the N, the next three for the denominator and the last three for the percentage. To make it a bit more human-readable we can remove the repeated information and read it like this:

ARMID	QualifierID	Role	Variable	ValueNum	ValueCode	ValueDecode
DM_SEX	1	ByVariable	TRTAN:TRTA	1		Treatment A
DM_SEX	1	ByVariable	SEXN:SEX	1	M	
DM_SEX	1.01	Statistic		25	N	
DM_SEX	1.02	Statistic		49	Denom	
DM_SEX	1.03	Statistic		51.0204081633	Percent	

Or even transpose it and add the other results (though this is not planned as deliverable) where the first row corresponds with the nine records from the previous example:

QualifierID	TRTAN	TRTA	SEXN	SEX	N	Denom	Percent
1	1	Treatment A	1	M	25	49	51.0204081633
2	1	Treatment A	2	F	24	49	48.9795918367
3	2	Treatment B	1	M	25	51	49.0196078431
4	2	Treatment B	2	F	26	51	50.9803921568
5	9	Total	1	M	50	100	50
6	9	Total	2	F	50	100	50

ANALYSIS OUTPUT METADATA

The AOM is still a work in progress, so no specific examples can be given for this yet. It will build upon the metadata previously discussed, including the analysis concepts. If the Actual Treatment concept is used in the analysis, then we can use the same concept as column variables in a table, which means that the columns will be sorted by TRTAN and labeled using the TRT codelist. Similarly, if the analysis was done on the Age at First Dose concept then in AOM we can define that the corresponding unit needs to be printed somewhere without specifying that the unit is stored in AAGE1U.

AOM defines an output. As mentioned before this can be an end-of-text table, figure or listing, but it can also be a machine-readable file to be consumed by another system. It defines which sets of ARD should be included, how to combine them and how to report them. Similar to ARM, the AOM is tool agnostic meaning it won't contain code or pseudo-code but can contain pre-defined methods that are translated into actionable code by the tools. An example of this is more complex sorting like sorting an adverse events table by descending frequency for system organ class and within that by descending frequency for preferred term. This would be specified with a pre-defined method as opposed to pseudo-code or trying to force it into an ADaM variable.

For a table the AOM will be split into two parts: transformation AOM to transform the ARD into reportable data and reporting AOM to actually report the data. The transformation AOM will include information on:

- Which sets of ARD to include and how to combine them
- Whether any subsetting needs to be done, for instance not all statistics generated might have to be printed in the table

- How the output should be sorted, which can be based on specific variables or more complex sorting through pre-defined methods
- How values should be combined, formatted, and aligned
- How rows and groups of data should be structured in the output

Where transformation AOM applies to a section in an output (e.g., the sex analysis in the demography table or the age analysis in the same table), reporting AOM applies to the entire output and will include:

- Titles and footnotes
- By group processing and respective subtitles
- The structure of columns and labels including their sorting
- Column headers and spanning headers
- The printing of big N in row labels, column headers, spanning headers, or subtitles

NEXT STEPS

Since this is very much a work in progress there are still a lot of things to do before we have fully implemented our vision. Our priority is to finalize the ARM and ARD structures. In addition, when producing ARD on numeric variables the precision of the input should be stored so that for AOM we can define what precision statistics should be reported in relative to the input precision (e.g. printing mean with one decimal more than the input). We also want to be able to overrule this precision.

The next step for ARM will be looking into non-trivial statistics. Most of our analyses only use frequency counts or summary statistics, but we also want to do more complex statistics using ARM and AOM. This will require updates to the ARM structure at least and more methods to be defined.

The structure of AOM needs to be further defined for tables and in addition figures and listings need to be added. For listings no ARM and ARD will be produced, but we need some form of AOM to define what ADaM data goes into a listing and how. In addition, the AOM for machine-readable outputs need to be defined. Examples of machine-readable outputs are:

- XML files for disclosure of results on clinicaltrials.gov
- XML files for disclosure of results on EudraCT
- JSON files to be consumed by Yseop, an AI-based, natural language generation platform for clinical study report automation

We also want to use the metadata even more. For instance, for an adverse event table that uses the *Actual Treatment* and *System Organ Class, Preferred Term* concepts as by variables, the *Treatment Emergent* and *Serious Adverse Event* range checks and the *Safety Analysis Set* concept for the analysis set we have all information to build the title of the output automatically instead of defining it manually.

A big topic is of course the cultural change and change management. This moves us away from programming from scratch or using macros or packages in SAS and R towards defining analyses and outputs in metadata, which is a fundamental change in how programmers work.

Tool development is currently ongoing in R. Tools for ingesting ARM and producing ARD and for ingesting ARD and AOM to produce tables are progressing nicely. The production of tables will be based on the open-source R package tfrmt which is being developed by GSK and is available on both github and CRAN.

Despite our experimentation starting with metadata in Excel we want the users to work with the metadata in an MDR. Obviously, there is no off-the-shelf solution, so we're investigating what MDR and/or GUI we need and where to build it in.

We're using what we call Analysis Concepts, which in essence are related to Biomedical Concepts (BC). We will continue to monitor the developments in the industry on BCs and where relevant adapt to include them.

Lastly, the goal is to link all our standards and study deliverables through a linked data model. We're starting with linking SDTM and ADaM, but the goal is to be able to link everything from protocol to outputs and CSR (protocol – eCRF – SDTM – ADaM – ARM – ARD – AOM – outputs – CSR).

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Edwin van Stein
GSK
Email: edwin.j.stein@gsk.com
Web: https://linkedin.com/in/ejvanstein

Brand and product names are trademarks of their respective companies.