Paper SI02

Automating ADaM Creation in Clinical Trials Part 2

Lyubov Sushchenko, Sanofi, Bridgewater, USA
Rie Ichihashi, Sanofi, Tokyo, Japan

ABSTRACT

The automation of ADaM creation in clinical trials has become a focal point within the pharmaceutical industry. Building on our presentation at PHUSE US Connect 2023 [1], where we introduced Sanofi's initial approach to ADaM/SDTM automation, we are excited to present the continued evolution of this project. As we advanced in the development of the user interface and tested pseudo-codes in pilot studies, we discovered significant support for the creation of value-level metadata (VLM) pseudo-codes. An unexpected yet valuable outcome of this automation initiative has been the reorganization of internal template programs, derivations, and macros, leading to a more streamlined automation process. In this presentation, we will again provide concrete examples of pseudo-code to illustrate its capacity to generate clear, specific, consistent, and unambiguous derivation rules, facilitating seamless automation. We will explore the end-to-end flow and discuss the degree of automation achieved. Emphasizing the importance of defining and developing standard specifications, we highlight how automation enhances data quality, reduces programming time, and simplifies routine tasks, echoing the principles outlined in our previous paper.

Keywords: standards, automation, pseudo-code, specification, derivation, transformation, VLM, ADaM

INTRODUCTION

The introduction of metadata-assisted programming represents an innovative approach to the automation of value-level metadata (VLM) in clinical trials. This methodology leverages the power of metadata to streamline and enhance the programming process. By incorporating metadata assistance, we not only expedite the programming workflow but also ensure a higher level of accuracy and consistency in the derivation of values. This not only aligns with industry standards but also paves the way for increased efficiency, reduced errors, and improved traceability, ultimately contributing to the overall quality and reliability of clinical trial data. Embracing metadata-assisted programming signifies a forward-looking commitment to precision and excellence in advancing the methodologies that underpin success on the study level. Within the context of this paper, we will explain standard derivations presented in pseudo-code form in VLM metadata.

ADAM VLM OPTIMIZATION AND THE PSEUDO-CODE

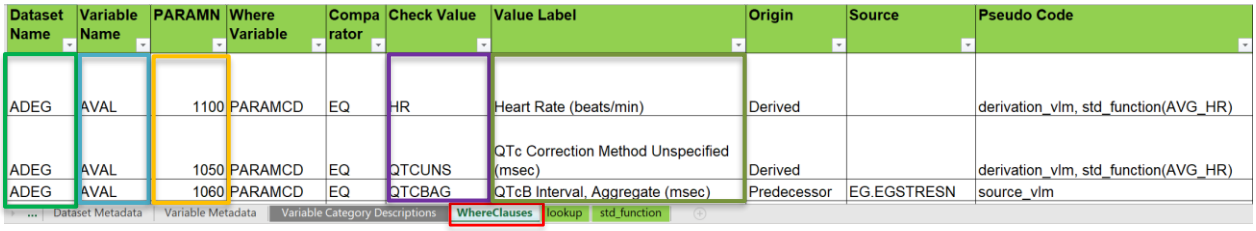
In our earlier publication, we presented a distinctive approach to formulating specifications and pseudo-codes that define variable metadata. In the context of VLM, we adhere to these principles as well, while tailoring them to conform to the VLM structure and streamlining the associated process. Pseudo-codes in VLM differ from Variable Metadata, yet there exists a relationship between the two. In VLM, the derivation of variable values varies based on conditions and the involvement of one or more variables. A good example is the derivation of AVAL/AVALC in BDS datasets, which differs depending on the values of PARAM/PARAMCD. **Figures 1 and 2** illustrate the linkage between Variable metadata and Value-level metadata.

Figure 1. WhereClauses pseudo-code in Variable Metadata

Dataset Name	Relative Order	Variable Name	Standard Length	Controlled Terminology	Derivation	Assigned	Pseudo Code
ADEG	1120	PARAM	200	EGPARM		Assigned as "[At a minimum EG.EGTEST concatenated with EG.EGSTRESU or derived parameter name]"	WhereClauses
ADEG	1130	PARAMCD	8	EGPARMCD		Assigned as "[A short name assigned to PARAM]"	WhereClauses
ADEG	1140	PARAMN	8	[EGPARMN]		Numeric code for PARAM	WhereClauses
ADEG	1170	AVAL	8		[EG.EGSTRESN or imputed/derived analysis value]		WhereClauses

... | Dataset Metadata | Variable Metadata | Variable Category Descriptions | WhereClauses | lookup | std_function

Figure 2. VLM example with pseudo-code column

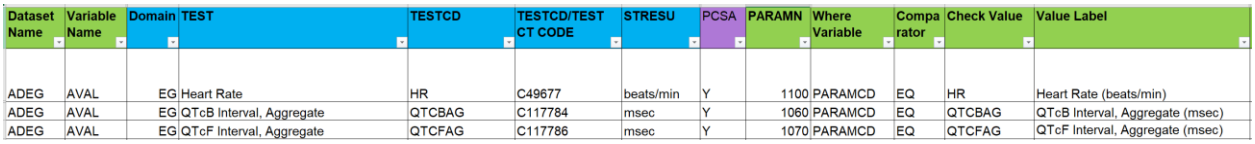


Dataset Name	Variable Name	PARAMN	Where Variable	Comparator	Check Value	Value Label	Origin	Source	Pseudo Code
ADEG	AVAL	1100	PARAMCD	EQ	HR	Heart Rate (beats/min)	Derived		derivation_vlm, std_function(AVG_HR)
ADEG	AVAL	1050	PARAMCD	EQ	QTCUNS	QTc Correction Method Unspecified (msec)	Derived		derivation_vlm, std_function(AVG_HR)
ADEG	AVAL	1060	PARAMCD	EQ	QTCBAG	QTcB Interval, Aggregate (msec)	Predecessor	EG.EGSTRESN	source_vlm

Dataset Metadata | Variable Metadata | Variable Category Descriptions | WhereClauses | lookup | std_function

Moreover, we have integrated a reference table for data collected in the study SDTM. To alleviate the tedium of data entry, a metadata-assisted programming interface will automatically pre-select ADaM VLM based on the existing SDTM data, as illustrated in **Figure 3**. This strategy enhances the traceability and visibility of the data during the dataset development phase. These enhancements aim to improve automation and alleviate the necessity for error-prone manual tasks.

Figure 3. VLM example with reference table



Dataset Name	Variable Name	Domain	TEST	TESTCD	TESTCD/TEST CT CODE	STRESU	PCSA	PARAMN	Where Variable	Comparator	Check Value	Value Label
ADEG	AVAL	EG	Heart Rate	HR	C49677	beats/min	Y	1100	PARAMCD	EQ	HR	Heart Rate (beats/min)
ADEG	AVAL	EG	QTcB Interval, Aggregate	QTCBAG	C117784	msec	Y	1060	PARAMCD	EQ	QTCBAG	QTcB Interval, Aggregate (msec)
ADEG	AVAL	EG	QTcF Interval, Aggregate	QTCFAG	C117786	msec	Y	1070	PARAMCD	EQ	QTCFAG	QTcF Interval, Aggregate (msec)

Due to the relatively simpler nature of VLM metadata compared to Variable metadata, only a limited number of pseudo-codes have been identified. Presented below are their respective definitions, refer to **Table 1** below

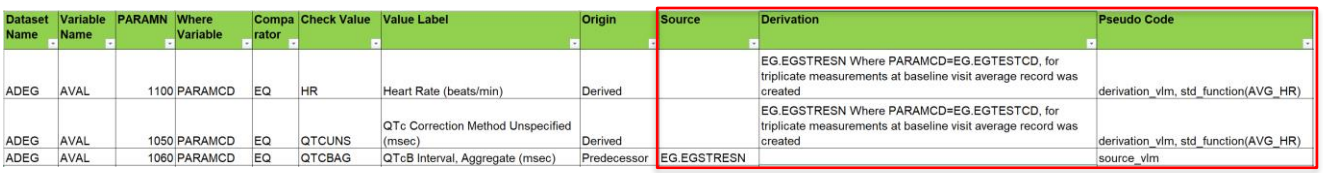
Table 1. VLM Pseudo-codes

Pseudo Code	Description
source_vlm	Indicates the variable value has an origin of “Predecessor” where the source variable is defined in the “Source” column. This is used mainly for SDTM predecessor variable values, retained by ADaM variables
derivation_vlm	Indicates that detailed derivation specs are available in the “Derivation” column in the “WhereClauses” tab
std_function	Complex derivations that involve multiple datasets and/or multiple steps of computations. Mostly processed by the macros or by the user-developed functions

The user interface for metadata-assisted programming is currently undergoing active development, with various automation functions in progress. Nonetheless, for this paper, we employ Excel to exemplify the creation of the pseudo-code. Pseudo-codes can be generated alongside Variable metadata when dealing with straightforward derivation rules.

For example, **Figure 4**, shows a simple pseudo-code “derivation_vlm” and “source_vlm”. These pseudo-codes use the corresponding variable metadata elements directly. Thus, the information in columns ‘Source’ and ‘Derivation’ is used by the pseudo-code is used to pass the key element to the code generator.

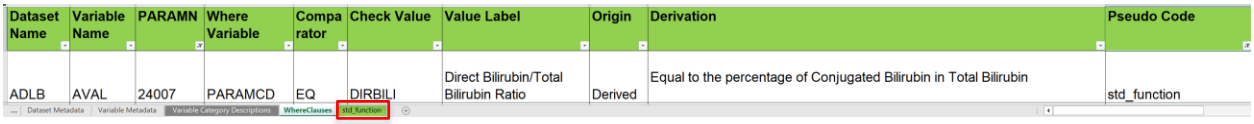
Figure 4. Columns for additional information



Dataset Name	Variable Name	PARAMN	Where Variable	Comparator	Check Value	Value Label	Origin	Source	Derivation	Pseudo Code
ADEG	AVAL	1100	PARAMCD	EQ	HR	Heart Rate (beats/min)	Derived		EG.EGSTRESN Where PARAMCD=EG.EGTESTCD, for triplicate measurements at baseline visit average record was created	derivation_vlm, std_function(AVG_HR)
ADEG	AVAL	1050	PARAMCD	EQ	QTCUNS	QTc Correction Method Unspecified (msec)	Derived		EG.EGSTRESN Where PARAMCD=EG.EGTESTCD, for triplicate measurements at baseline visit average record was created	derivation_vlm, std_function(AVG_HR)
ADEG	AVAL	1060	PARAMCD	EQ	QTCBAG	QTcB Interval, Aggregate (msec)	Predecessor	EG.EGSTRESN		source_vlm

On the other hand, when the pseudo-code is “std_function”, additional information needs to be entered in another tab that has the same name as the pseudo-code, see **Figure 5**.

Figure 5. Tabs for additional information



Dataset Name	Variable Name	PARAMN	Where Variable	Comparator	Check Value	Value Label	Origin	Derivation	Pseudo Code
ADLB	AVAL	24007	PARAMCD	EQ	DIRBILI	Direct Bilirubin/Total Bilirubin Ratio	Derived	Equal to the percentage of Conjugated Bilirubin in Total Bilirubin	std_function

Dataset Metadata | Variable Metadata | Variable Category Descriptions | WhereClauses | std_function

LET US DISCUSS VLM DERIVATION TYPES IN MORE DETAIL

1. Copy a variable value from the variable in the main source dataset (pseudo-code= “source_vlm”). See **Figure 5**. This is used mainly for predecessor values retained from SDTM. Information is parsed into program code to keep the variable value from the main source dataset and used as a source for define.xml to display the predecessor’s variable value. The main source dataset is the dataset that includes most of the source variables for that ADaM dataset (e.g., SDTM EG for ADEG).

Examples: ADEG.AVAL retains values from EG.EGSTRESN or ADLB.AVALC retains values from LB.LBSTRESC, etc.

Figure 6. Pseudo-code “source_vlm” in the WhereClauses tab

Variable Name	PARAMN	Where Variable	Comparator	Check Value	Value Label	Origin	Source	Pseudo Code
AVAL	1060	PARAMCD	EQ	QTCBAG	QTcB Interval, Aggregate (msec)	Predecessor	EG.EGSTRESN	source_vlm
AVAL	1070	PARAMCD	EQ	QTCFAG	QTcF Interval, Aggregate (msec)	Predecessor	EG.EGSTRESN	source_vlm
AVAL	1080	PARAMCD	EQ	QTCN	QTcN - Pop. Specific Correction Formula (msec)	Predecessor	EG.EGSTRESN	source_vlm

The rendered SAS ® code is as follows:

```
AVAL=EGSTRESN;
```

2. Simple computation which appears in define.xml as a computational algorithm is directly used as the program code (pseudo-code = “derivation_vlm”). See **Figure 7**.

For example, four basic arithmetic operations: addition, subtraction, multiplication, and division. The source variable(s) value(s) should be available in the same record in the main source dataset.

Example: ADES.AVAL, ADES.PARAMCD

Figure 7. Pseudo-code “derivation_vlm” in the WhereClauses tab

Dataset Name	Variable Name	Where Variable	Comparator	Check Value	Value Label	Origin	Derivation Type	Derivation	Pseudo Code
ADES	AVAL	PARAMCD	EQ	DURD	Total Treatment Duration in (Days)	Derived	Computation	ADSL.TRTEDT - ADSL.TRTSDT +1	derivation_vlm

The rendered SAS ® code is as follows:

```
If PARAMCD=”DURD” then AVAL=TRTEDT-TRTSDT+1;
```

3. Deriving more complicated VLM’s (pseudo-code = “std_function”). See **Figure 8**.
This is used when deriving VLM needs to perform multiple data manipulations, for example, calculating an average record for multiple measurements.

Example: ADEG.AVAL, ADEG.DTYPE, ADEG.PARAMCD, if the patient had multiple assessments on the same date, then calculate additional AVAL average record, and assign DTYPE value.

Figure 8. Pseudo-code "std_function" in the WhereClauses/std_function tabs

Dataset Name	Variable Name	PARAMN	Where Variable	Comparator	Check Value	Value Label	Origin	Derivation	Pseudo Code
ADEG	AVAL	1100	PARAMCD	EQ	HR	Heart Rate (beats/min)	Derived	EG.EGSTRESN Where PARAMCD=EG.EGTESTCD, for triplicate measurements at given visit average record was created	derivation_vlm, std_function(AVG_HR)

The programmer passes derivation information to the function AVG_HR in "std_function" tab:

Dataset Name	Variable Name	SAS Macro	Std function reference	Parameter Name	Parameter Value
ADEG	AVAL	dtype_average	AVG_HR	data_in	##dsin##
ADEG	AVAL	dtype_average	AVG_HR	data_out	##dsout##
ADEG	AVAL	dtype_average	AVG_HR	filter_datain	PARAMCD in ("HR")
ADEG	AVAL	dtype_average	AVG_HR	summarized_var	AVAL
ADEG	AVAL	dtype_average	AVG_HR	subj_key	STUDYID USUBJID
ADEG	AVAL	dtype_average	AVG_HR	groupby_var	ADT
ADEG	AVAL	dtype_average	AVG_HR	adt_first_last	first
ADEG	AVAL	dtype_average	AVG_HR	adtm_first_last	first
ADEG	AVAL	dtype_average	AVG_HR	atm_first_last	first

... | Dataset Metadata | Variable Metadata | Variable Category Descriptions | WhereClauses | lookup | std_function

The rendered SAS ® code is as follows:

```
*** Create average records ***;
proc sql;
    create table ADEG_01_result
    as select distinct STUDYID, USUBJID
    , PARAMCD, PARAM, ADT, mean(AVAL) as AVAL
    , "AVERAGE" as DTYPE
    , min(ADT) as ADT
    , min(ADTM) as ADTM
    , min(ATM) as ATM
    from ADEG_01(where= (PARAMCD in ("HR")))
    group by STUDYID, USUBJID, ADT;
quit;

*** Set back the derived records with original records ***;
data ADEG_02;
   set ADEG_01 ADEG_01_result;
run;
```

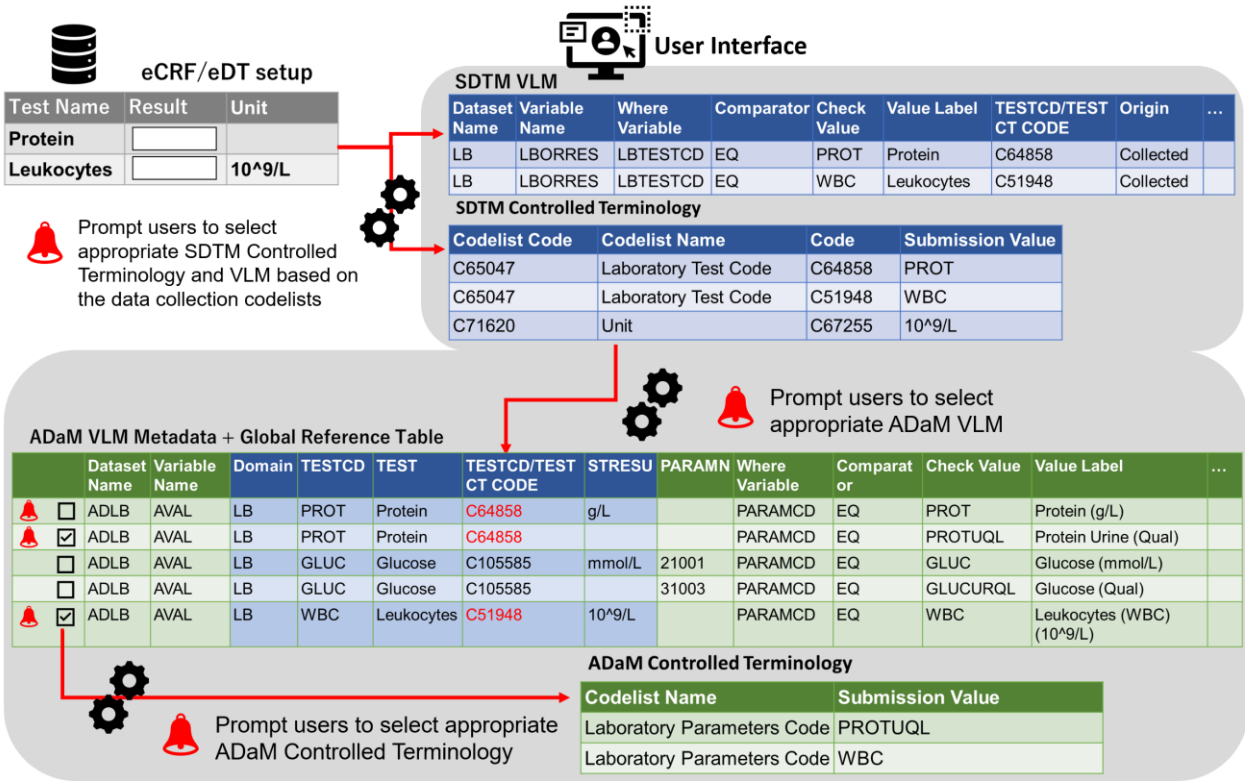
Pseudo-codes for each VLM can be defined at the global standard level, allowing users to customize these pseudo-codes based on the requirements of the study.

GENERAL PRINCIPLES OF VLM METADATA AUTOMATION

Populating VLM metadata at the study level can be a challenging task for everyday programmers, and there's a common tendency to postpone it until the end of the study or until submission is confirmed. This approach may lead to high levels of anxiety during the Data Base Lock (DBL) and post-DBL processing, especially since it's intertwined with CT (Controlled Terminology) selection for the study.

Automating the labor-intensive Value Level Metadata (VLM) creation poses a considerable challenge. Our metadata solution facilitates the connection from the eCRF/eDT to SDTM and finally to ADaM, streamlining metadata management and simplifying the generation of the define.xml file. The one of goals of the user interface is to automatically identify all parameters collected in SDTM and prompt the user to select the relevant ones for analysis. Please, refer to the visualized data flow automation below in **Figure 9**.

Figure 9: Data flow from eCRF/eDT to SDTM and ADaM



Beginning with the setup of electronic Case Report Forms (eCRF) and electronic Data Capture (eDT) during the study initiation, the interface facilitates the definition of ADaM VLM parameters through seamless integration with a comprehensive SDTM reference table. Additionally, it supports the derivation of specific values for ADaM dataset variables such as PARAM, PARAMCD, PARAMLy, AVAL, and AVALC, thereby optimizing the overall efficiency of the process for the collected data. Derived ADaM VLM parameters are referencing ADaM CT values directly.

The reference table is established at the global metadata level, creating a direct link with CT values. Any modifications or updates to the controlled terminology are automatically identified by the user interface during global metadata updates, ensuring that the parameter reference table remains consistently up to date.

BENEFITS OF VLM AUTOMATION

This proposed approach for the automation of VLM metadata addresses the challenges associated with traditional manual methods. Here's an overview of the benefits proposed automation can offer:

- **Early Planning**: Linking SDTM and ADaM metadata and pre-selecting collected parameters provides a systematic way to encourage programmers to incorporate VLM metadata population into the early stages of the study. This integration ensures that the VLM metadata is considered alongside other critical metadata elements from the beginning, promoting a more holistic approach to study development.

- **Continuous Monitoring**: The proposed user interface that warns study programmers of changes in SDTM during the study conduct is a proactive measure. This feature helps in continuously monitoring the status of the metadata throughout the study, allowing for real-time adjustments as needed. It prevents surprises at later stages by keeping programmers informed of any changes that might impact the VLM metadata.

- **Feedback Loop and Agile Development**: Establishing a feedback loop for programmers to communicate challenges or issues related to VLM metadata population is a key component. This feedback loop will contribute to an agile development strategy, allowing for iterative improvements in the automation process based on real-world challenges faced by programmers. Continuous improvement is essential for adapting to the evolving needs of studies.

By incorporating these features into the automation of VLM metadata population, we are not only addressing the immediate challenges but also promoting a more efficient and adaptive approach to study management. This aligns well with industry best practices, enhances collaboration between different stages of the study, and ultimately contributes to the success of the overall clinical trial process.

CONCLUSION

In conclusion, the journey of automating ADaM creation in clinical trials, as presented in this paper, underscores a significant evolution in the pharmaceutical industry's approach to data processing. The continuous development of the user interface and the exploration of pseudo-codes in pilot studies have not only led to substantial support for value-level metadata (VLM) pseudo-codes but have also unexpectedly streamlined internal processes, including template programs and derivations. This not only demonstrates the adaptability of the automation initiative but also reflects a commitment to enhancing efficiency. Concrete examples of pseudo-code prove its capacity to generate precise and consistent derivation rules, contributing to seamless automation. The emphasis on defining and developing standard specifications echoes principles outlined in previous work, reinforcing the importance of these standards. The innovative introduction of metadata-assisted programming, particularly in the context of VLM, showcases a forward-looking commitment to precision and excellence in clinical trial methodologies. The optimization of ADaM VLM, illustrated through pseudo-codes and a reference table, contributes to improved traceability and visibility during dataset development. The proposed automation approach addresses challenges associated with manual methods, offering benefits such as early planning, continuous monitoring, and a feedback loop for agile development. This comprehensive and adaptive approach aligns with industry best practices, fostering collaboration across different stages of the study and ultimately contributing to the success of a clinical study processing.

ACKNOWLEDGMENTS

The authors of this paper would like to express their gratitude to the Sanofi ART team, for their support and ideas, especially Faisal Mohammed. ADaM MAP team for their dedication and effort. B&P programmers for providing feedback and volunteering to test out new tools. And Beilei Xu for inspiring us to write this paper.

REFERENCES

[1] L. Sushchenko, R. Ichihashi, “Automating SDTM and ADaM Creation in Clinical Trials”, PHUSE US Connect 2023

CONTACT INFORMATION

Your comments and questions are valued and encouraged.

Contact the authors at:

Author Name: Lyubov Sushchenko
Company: Sanofi US
Address: 55 Corporate Drive
City / Postcode: Bridgewater, 08807
Email: Lyubov.Sushchenko@sanofi.com
Web: https://www.sanofi.com

Author Name: Rie Ichihashi
Company: Sanofi K.K.
Address: Tokyo Opera City Tower, 3-20-3, Nishi Shinjuku
City / Postcode: Tokyo 163-1488, Japan
Email: Rie.Ichihashi@sanofi.com
Web: https://www.sanofi.com