PhUSE 2017

Paper DS11

Consuming your metadata for a healthy mapping

Alan Cantrell, PAREXEL International Limited, Sheffield, England
Julius Kusserow, PAREXEL International Limited, Berlin, Germany

ABSTRACT

During 2016, PAREXEL implemented a pattern based Metadata Repository (MDR) using a hub and spoke design with hubs containing generic variables and spokes containing standard specific variables (e.g. SDTM and ADaM).

With the spokes created along with connections to the domain definitions (Hub); the metadata can be extracted from the MDR and consumed by SAS program code along with the data it describes.

This brings several benefits

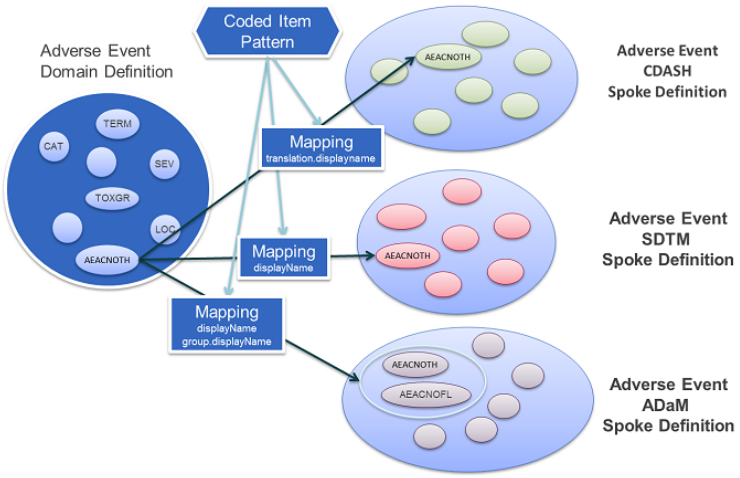
- Mapping between the spokes is performed in a consistent way
- Time saved by autogenerated non-study specific code to create SDTM and ADaM datasets
- Standard metadata and code maintained in controlled manner
- Programmers can begin the SDTM and ADaM programming work before data received
- Work can begin on ADaM dataset programming directly from raw for pattern-based variables

We will demonstrate implementation by looking at the creation of domain definition variables, creation of spoke definition variables; linkage definition between the Hub and Spokes, extraction of metadata and consumption by SAS programs to generate tabulation and analysis datasets.

INTRODUCTION

During 2016, the PAREXEL MDR was implemented to define, store and manage our metadata for data collection, tabulation and analysis; this metadata was connected by the use of pattern metadata.

- Domains are defined through a hub & spoke. Each domain definition within the hub contains the specification of the variables relevant for the domain, combined into semantically consistent groups (e.g. Value and Unit are linked together). Hubs are agnostic of any standards. Each hub is linked to several spokes, such as data collection (CDASH), data tabulation (SDTM) and analysis (ADaM) that includes the variables or variable group for that specific spoke, linked to the hub.



- Mapping through the hub and spokes is defined by patterns. The patterns specify how a variable/variable group with a certain data type in the hub can be mapped with a variable/variable group in the different spokes. For instance the pattern Coded Item indicates that a variable of the type coded concept in the hub is mapped to a single variable in CDASH and SDTM and with many different variables in ADaM. We identified 20+ patterns that govern all CDISC standards.

With the collection, tabulation and analysis metadata (Spokes) in place along with the connections to the domain definitions (Hub); the metadata can be extracted from the MDR and consumed by SAS program code along with the data it is describing.

This brings several benefits

- Mapping between the spokes is performed in a consistent way
- Time saved by not writing study specific code to create SDTM and ADaM datasets
- Standard metadata and code maintained in controlled manner
- Programmers can begin the SDTM and ADaM programming work before data received

During the presentation, we will demonstrate how we have implemented an example set of pattern based metadata and together with the study data; how this can be consumed by SAS programs in a statistical computing environment (SCE) to generate SDTM and ADaM data. We will look at the creation of a domain definition (Hub) variable, the creation of spoke definition variables; the linkage definition between the Hub and Spokes; the extraction of this metadata; generation of SAS code and consumption of the metadata by these SAS programs to generate tabulation and analysis datasets.

USING THE METADATA

In last year's paper we introduced the idea of patterns to facilitate the recreation of linear mapping using complex datatypes.

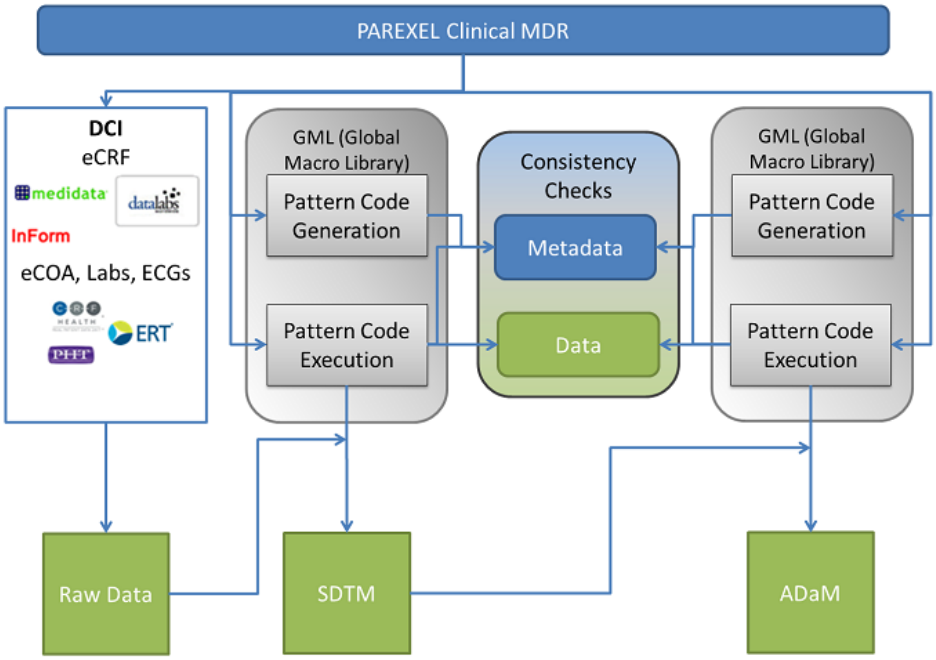
This metadata then forms a part of the study instance metadata (SIM) that is the metadata which together defines a study definition containing database design metadata such as forms, controlled terminology; tabulation metadata (such as SDTM) and analysis metadata (such as ADaM).

The key aspect here is to take this metadata and use it to facilitate the creation of 3 distinct outputs:

- Raw data (that being the data that is collected within an eCRF (electronic case report form) or vendor collected data (such as laboratory, electrocardiogram, questionnaires)
- Tabulation data (such as CDISC SDTM)
- Analysis data (such as CDISC ADaM)

To create each of these three distinct outputs we need to look at how the metadata from the PAREXEL Clinical MDR can be used.

The figure below shows the PAREXEL Clinical MDR structure at the top and how through various steps (described below); output data can be produced.



GENERATING PATTERN CODE

For each pattern that is defined within the MDR; there needs to be a corresponding algorithm to translate the pattern metadata connectivity and input data into output data. This algorithm needs to be generated based on the pattern used which is identified by the construct of the pattern from the MDR.

EXECUTING PATTERN CODE

Once the pattern code is generated in the previous step; the code is executed together with the input data (this would be raw data for SDTM and SDTM data for ADaM). If the code executes successfully; output data is created (e.g. SDTM or ADaM) based on the pattern code used.

CONSISTENCY CHECKING AND GOVERNANCE

Although output may be produced without any code execution issues; it is important that checks are performed to ensure consistency. Consistency checks are performed at both the pattern code generation stage as well as the pattern code execution stage.

During the pattern code generation stage; consistency checks are performed on the metadata to check that each variable in the metadata is used

During the pattern code execution stage; consistency checks are performed on both the metadata and data to check that the individual metadata is consistent between the study instance metadata from the MDR; the metadata in the pattern code generated and the metadata (structure) and values (content) from the raw data.

A WORKED EXAMPLE

In the example; we will look at the variable AEACNOTH; this variable is commonly used in clinical trials to collect other action taken (not study medication related) due to an adverse event occurring.

The expected responses in the raw data could be:

- None
- Concomitant medication
- Hospitalization or prolongation of hospitalization
- Therapeutic or diagnostic procedure

A pattern “Coded Item” is attached to the variable.

This pattern allows a number of elements to be collected:

- displayName – containing the text which is submitted in the tabulation dataset variable AEACNOTH
- translation.displayName – containing a translation of the values expected in the tabulation dataset variable AEACNOTH, e.g. this may be similar values to be presented on the CRF for investigator use
- group.displayName – used to define the grouping of values
- controlled terminology – containing the input and decoded output value mappings (e.g. 0 = “None”)

Based on the elements that form the pattern and the relationship between the elements; a series of SAS code is generated to translate the metadata received (e.g. the pattern set-up) to populate each element’s content based on some input value. In this particular case; SAS code is created to generically populate each element of a “Coded Item” pattern. In this way, the code will not only work for AEACNOTH but also for any other variables which have a “Coded Item” pattern associated with them.

As part of this code; consistency checks will also be created to check the metadata received for the “Coded Item” pattern is compliant with the metadata expected for the pattern; that is the PAREXEL Clinical MDR metadata meets the program code expectations. The purpose of this check is to ensure that if the metadata was changed to be non-compliant then the related SAS program is flagged for a consistency check.

Once the pattern code has been generated; it can be executed once the clinical data is available. In the case of tabulation dataset population; the raw data from the data collection instruments is read into the pattern code as the source data along with the metadata from the PAREXEL Clinical MDR.

In addition to the generated code; “free code” can be added; this is to allow the flexibility of any additional code being added that does not comply with the standard macro but is needed for an exception. Free code will be limited and also assessed if it can be added into the standard pattern generation code. The free code is added at the generation stage rather than the execution stage to ensure the code is maintained in a central place and also to allow consistency checks to be performed to ensure that the free code has not caused non-compliance.

The data is checked for consistency to the metadata to ensure that the values of the data are expected. For example if an input value of “5” was existing in the raw data, and this was not expected per the related metadata; a consistency check will fire to highlight this.

Once the pattern code has been executed and the consistency checks pass; the output variable AEACNOTH is populated in the SDTM AE dataset.

Other variables in the AE dataset will follow the same approach as above; some variables will use the “Coded Item” pattern code; others will use different pattern code.
Further to this; once the AE dataset is created; similar pattern code would be executed for other events datasets.

In addition; the ADaM variables (in the ADAE dataset); AEACNOTH and AEACNOFL can be populated. AEACNOTH will follow the same rules as SDTM (AE.AEACNOTH) as the variable is the same and as such the metadata in the pattern is also (displayName). AEACNOFL uses group.displayName which consists of a set of controlled terminology mapping the raw values to values of “N” or “Y”; for example, an input value of “NONE” would be mapped to “N”; while other values (where action was taken) would be mapped to “Y”. The advantage here is that the ADaM program generation can be produced without the SDTM data. However the ADaM code execution will use the SDTM data as the input data for mapping.

CONCLUSION

By defining algorithms that relate to each pattern applied in the PAREXEL Clinical MDR; we can automate the majority of tabulation and analysis dataset mappings and associated population. The ability to create executable SAS code that can be applied consistently in multiple places ensures that mapping logic is consistent across both datasets but also projects rather than being manually defined on a project by project basis.
This provides a much improved efficiency in dataset population following the definition and set-up of each pattern.
The pattern code can also be created at the same time as the pattern definition in the MDR; this allows the definition of both elements in tandem and ensures that executable program code is available earlier in the project process.

REFERENCES

[1]. Effective use of a Metadata Repository across data operations: the need for a machine readable form of (part of) the protocol. PhUSE 2016. Isabelle de Zegher, Michele Gray, Mark Sullivan, Michael Goedde. www.phusewiki.org/docs/Conference 2016 DH Papers/DH01.pdf

[2]. E2E data standards, the need for a new generation of metadata repositories. PhUSE 2015. Isabelle de Zegher, Alan Cantrell, Julie James. www.phusewiki.org/docs/Conference 2015 DH Papers/DH04.pdf

[3]. Pattern based Metadata Repository: toward high quality data standards. PhUSE 2016. Alan Cantrell, Julius Kusserow, Julie James, Deb Copeland, Natraj Patro, Isabelle de Zegher. www.phusewiki.org/docs/Conference 2016 CD Papers/CD12.pdf

ACKNOWLEDGMENTS

The work presented in this paper is part of a wider initiative within PAREXEL around E2E Data Standards management, including the PAREXEL's Clinical MDR, a Statistical Computing Environment (SCE) supporting generation of SDTM, ADaM and TLFs, and a Data Consistency Checker ensuring that the data delivered from data collection instruments are delivered in the format defined in the MDR and expected by the SCE.

CONTACT INFORMATION

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

Alan Cantrell
PAREXEL International
1 South Quay Drive
Sheffield S2 5SU
Work Phone: +44 (0) 114 225 1351
Email: alan.cantrell@parexel.com
Web: https://www.parexel.com