PhUSE US Connect 2018

Paper SI18

Patterns emerging from chaos

Alan Cantrell, PAREXEL International Limited, Sheffield, England
Julius Kusserow, PAREXEL International GmbH, 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 standard specific variables such as SDTM and ADaM.

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

This brings several benefits

- Standard metadata and code is maintained in a controlled manner
- Programmers can begin the SDTM and ADaM programming work before real data is received

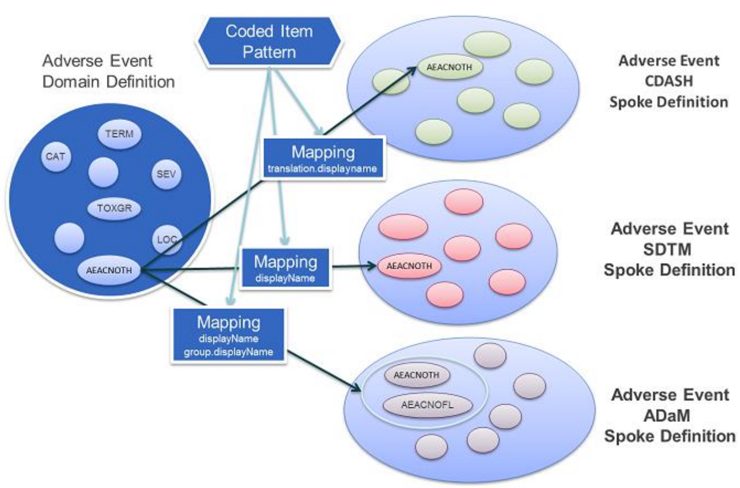
We will demonstrate how this model facilitates change to underlying metadata and the impact of change to downstream mappings. As underlying code is generated based on metadata; changing metadata will impact the code; however this needs governing in a controlled and efficient way rather than a chaotic uncontrolled fashion. Patterns and the associated code facilitate this controlled standards implementation.

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

We will demonstrate how this model facilitates change to underlying metadata and the impact of change to downstream mappings. As underlying code is generated based on metadata; changing metadata will impact the code; however this needs governing in a controlled and efficient way rather than a chaotic uncontrolled fashion. Patterns and the associated code facilitate this controlled standards implementation.

HISTORICAL CHANGES TO STRUCTURE

When programming is performed; a specification documenting the requirements and rules for mapping from a source to a target is created; this is based on the source structure and target structure being defined at a moment in time. In the case of an SDTM mapping specification; the source would be the database extract structure and content; the target would be the CDISC SDTM model.

Although the target structure would be unlikely to change on a project with a set CDISC SDTM version being followed; the source structure could be affected by a protocol amendment during the project life span; a change in expectations by the clinical trial sponsor due to an internal standards change.

As a CRO in a position to deliver multiple datasets / outputs on multiple projects for a sponsor; it is important to be in a position to react to these changes in an efficient manner ensuring consistency.

METADATA LINKAGE

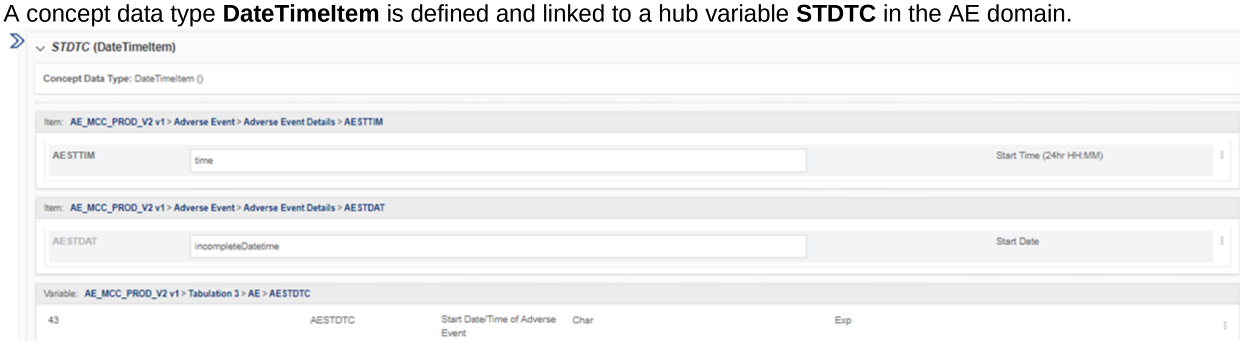
The **PatternDateItem** pattern facilitates the reading in of variable metadata “aeStDat_raw” with the attribute “item.ddmmmyyyy” – therefore expecting the input date to be of the format ddmmmyyyy, e.g. 25JAN2018. The pattern also facilitates the reading out of variable metadata “aeStDtC” with the attribute “item.yyyymmdd” – therefore outputting the date value in the format yyyymmdd, e.g. 20180125.

```
* Date variables;
** Start Date;
%gmPatternDateItem(   varsIn  =   aeStDat_raw = item.ddmmmyyyy
                    , varsOut =   aeStDtC     = item.yyyymmdd
                    , codeOut = startDateCode
                   );
```

Each pattern macro is called to generate SAS code in a consistent way; e.g. in this case; SAS code **startDateCode** is generated. Each of the generated SAS codes are then executed to facilitate the creation of an output such as the AE SDTM dataset.

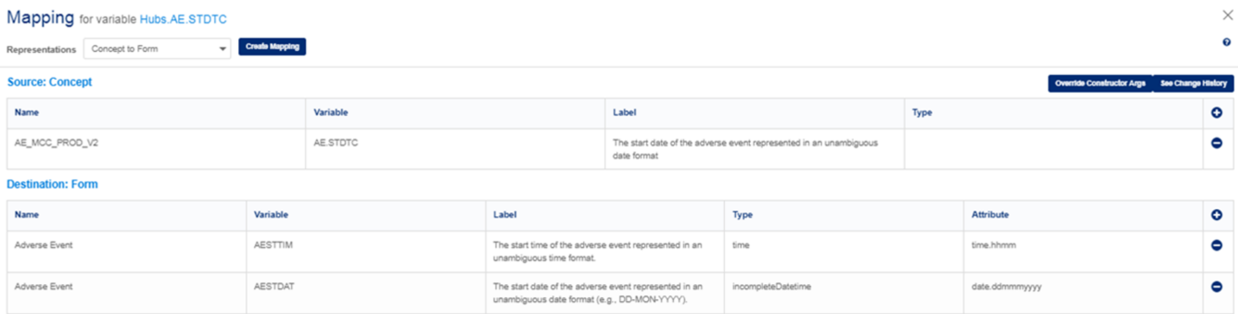
The pattern attributes are defined within the MDR and together form a concept data type.

A concept data type **DateTimeItem** is defined and linked to a hub variable **STDTC** in the AE domain.



The concept data type supports the collection and exchange of dates and times; including partial dates; e.g. when only year and month are known for a date and the day value is unknown.

Within the EDC adverse event form; the variables AESTDAT (AE Start Date) and AESTTIM (AE Start Time) are connected to the hub item AE.STDTC with attributes of date.ddmmmyyyy and time.hhmm respectively.


Mapping for variable Hubs.AE.STDTC

Representations: Concept to Form | Create Mapping

Source: Concept

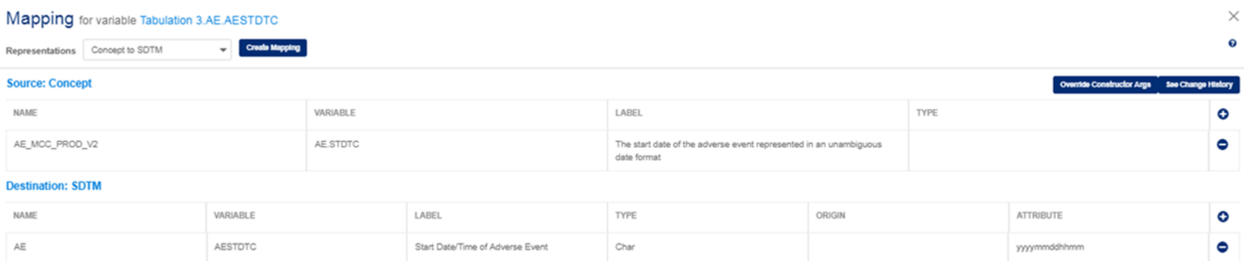
Override Constructor Args | See Change History

Name	Variable	Label	Type
AE_MCC_PROD_V2	AE.STDTC	The start date of the adverse event represented in an unambiguous date format	

Destination: Form

Name	Variable	Label	Type	Attribute
Adverse Event	AESTTIM	The start time of the adverse event represented in an unambiguous time format.	time	time.hhmm
Adverse Event	AESTDAT	The start date of the adverse event represented in an unambiguous date format (e.g., DD-MON-YYYY).	incompleteDatetime	date.ddmmmyyyy

Within the SDTM AE domain; the variable AE.AESTDTC (Start Date/Time of Adverse Event) is connected to the hub item AE.STDTC with an attribute of yyyymmddhhmm.


Mapping for variable Tabulation 3.AE.AESTDTC

Representations: Concept to SDTM | Create Mapping

Source: Concept

Override Constructor Args | See Change History

NAME	VARIABLE	LABEL	TYPE
AE_MCC_PROD_V2	AE.STDTC	The start date of the adverse event represented in an unambiguous date format	

Destination: SDTM

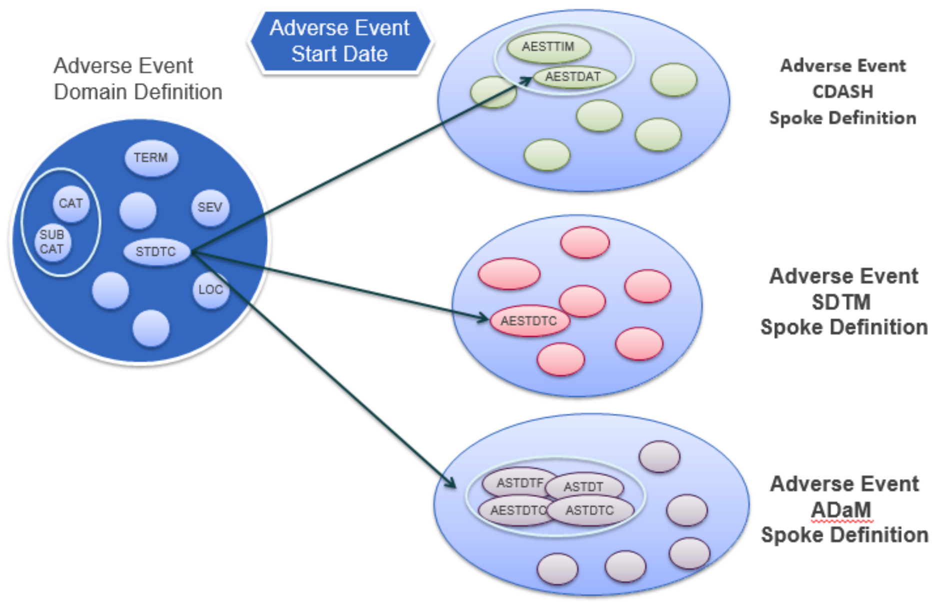
NAME	VARIABLE	LABEL	TYPE	ORIGIN	ATTRIBUTE
AE	AESTDTC	Start Date/Time of Adverse Event	Char		yyyymmddhhmm

Similarly the ADaM variables are also connected to the hub item and allows imputation to also be performed when the date collected in the form is partial. The analysis start date (perhaps imputed) and analysis start date imputation flag are also created as attributes.

```
* Date variables;
%gmPatternDateItem(   varsIn  =   aeStDtC = item.yyyymmdd
                    , varsOut =   aStDt = item.imputeFirst.sas
                                @ aStDtF = item.imputeFirst.flag
                    , codeOut = startDateCode
                   );
```

Similar to SDTM generation; each pattern macro is called to generate SAS code in a consistent way; e.g. in this case; SAS code **startDateCode** is generated. Each of the generated SAS codes are then executed to facilitate the creation of an output such as the ADAE ADaM dataset.

With this approach it ensures that the form items; SDTM and ADaM variables are all connected via a central concept data type with multiple attributes.



METADATA CHANGES AND IMPACT

As from above; the form items "Adverse Event Start Date" and "Adverse Event Start Time" and the SDTM variable "Start Date/Time of Adverse Event" are connected to a central variable AE.STDTC instead of being directly connected to each other in a linear way. Through this connection; the attributes for the form items can be changed while still ensuring the data produced in the SDTM is consistent.

With an attribute being changed such as the date collection format in the form; the connection to the hub variable AE.STDTC is still maintained. As this connection is still maintained and valid; it ensures that the connection from the hub to the SDTM variable AESTDTC is still able to be mapped from the updated form item.

Through this approach compared to the "linear" approach mapping directly from form item to SDTM variable and SDTM variable to ADaM variable; less manual intervention is required to adapt code in multiple places as it is generated based on the attributes selected in the study implementation.

GOVERNANCE AND PATTERNS

As the metadata connecting the hub and spokes (in this case EDC forms and SDTM datasets) is contained within the PAREXEL MDR and is version controlled when creating a study implementation; any updates (in accordance with protocol / sponsor change requests) made to the form items / form design are therefore also controlled.

As the SAS code used to generate and populate outputs such as SDTM and ADaM datasets is based on the corresponding version controlled metadata; this ensures that the SAS code has implicit governance.

CONCLUSION

By using patterns to connect the form items and SDTM variables to a central hub; we ensure that changes to the form design metadata on existing studies allow the SDTM mapping to remain in place. These updates are version controlled within a study implementation allowing the corresponding executable SAS code to be generated with the updated attributes in place. By generating the executable code based on the metadata rather than manually updating existing mapping programs; consistency and quality can be ensured and avoids making updates to existing study data mappings in an uncontrolled fashion.

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

[4]. Consuming your metadata for a healthy mapping. PhUSE 2017. Alan Cantrell, Julius Kusserow. www.phusewiki.org/docs/Conference 2017 DS Presentations NEW/DS11.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