PharmaSUG 2019 - Paper DS-308

Using CDISC Standards with an MDR for EDC to Submission Traceability

Paul Slagle, Syneos Health
Eric Larson, Syneos Health

ABSTRACT

CDISC Standards have long promised a way to add clarity to a submission through the use of traceability. This includes the ability of tracing the data collected in an EDC system through to the analysis provided to the regulatory authority. The challenge with doing this is that to manage all of the traceability is a documentation headache. With the use of a metadata repository, you can develop screens that can be pushed into an ODM compliant EDC system, such as Medidata Rave. From that push you can receive the raw data and transform that, using SAS, into a format compliant to SDTM standards while maintaining the connections of where the data came from in EDC. Adding Results Metadata, when entered into the metadata repository, then allows the creation of ADaM datasets which are built on SDTM data collected from the EDC system. This paper will demonstrate how an off the shelf MDR tool can be used with SAS to build these connections while maintaining compliance to CDISC standards.

INTRODUCTION

We all know CDISC is the standard for submitting data to regulatory agencies for drug approvals. Within CDISC, the critical models for submission are the Study Data Tabulation Model (SDTM) and the Analysis Data Model (ADaM). These models drive how data and derivations are stored in datasets and variables in much the same manner we have been creating using SAS®.

For years we have also talked about the use of data driven programming as the standard for building SAS programs. This concept endorses that the program uses data provided in the analysis to perform specific actions. It discourages the use of absolute values within the program and instead reads data from external files and datasets to deliver datasets and variables. Examples of data driven include the use of look up tables to map collected data to formatted data values.

So what if we combined the CDISC submission standards with the concepts of data driven programming to deliver submission data solely on the use of the specifications? This would reduce the redundant programming created to map collected data into SDTM while encouraging the use of that mapped data for data derivation, results generation, and finally creating output.

This paper explains a basic process for doing this which has been partially implemented within Syneos Health.

LET'S START WITH THE BASICS

For demonstrating the basics, let's focus first on just the specifications and how those are managed. Metadata is defined under ISO 11179 as data that is used for describing other data. The most basic part of metadata is the data element, which we reference as an element in this document for brevity. The basic attributes assigned to an element include information we are used to such as element type, labels, and lengths. In addition to the attributes we also assign characteristics which define how the element can be used such as for a variable, whether the element is approved, or specific therapeutic area the variable can be used for. Also of interest is the artifact of an element which shows how the variable was sourced.

Making metadata more interesting, elements can also be datasets or domains. In this case the characteristic of the element becomes an element set. By using this level of abstraction you can then maintain the attributes such as the label for the element set, assign whether the element set is approved or draft, and maintain the artifacts showing the source for the element set description.

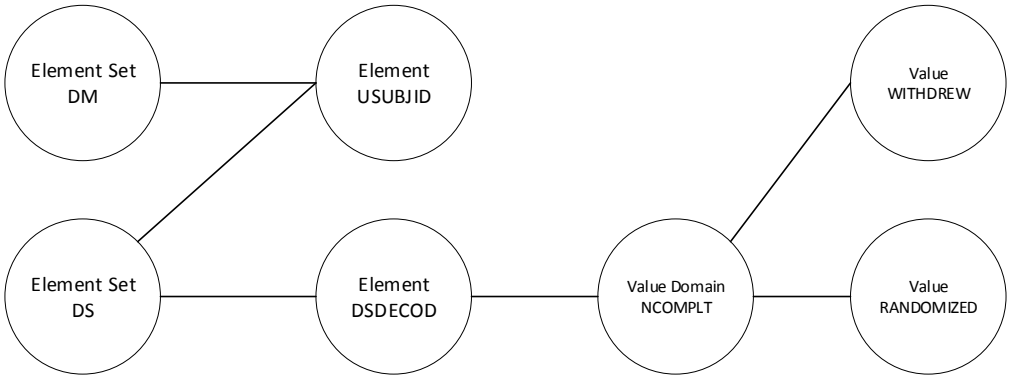
Once you have elements and element sets, the next piece in the metadata description is to show the relationships between elements and element sets. Relationships can be made in multiple directions depending on the need. As an example, an SDTM domain such as DM will have USUBJID as an element but USUBJID will have a relationship to DM, AE, CM, and many others. In this way you can define

USUBJID once while relating the element to the different element sets in which it is needed. These relationships are the important piece of the metadata standards to remember for this paper.

In order to provide completeness, we would be remiss to not include value domains as a different type of element in the specifications. Value domains describe specific values that can be associated with a specific attribute. Value domains will also have specific values as elements which can be related to the value domains. Because you have values, you can also describe the attributes for those and, in this way, describe value metadata for an element.

As you start to put the pieces of elements and relationships together, you begin to create diagrams as shown in Figure 1: One thing to remember in the creation of this abstraction is that the relationships need to be maintained as bidirectional. In that way you can trace not just which elements belong in a specific element set but then also which element sets contain the same elements.

Figure 1: Relationship structure of metadata



MANAGING CDISC STANDARDS WITH METADATA

Now that we have described the basics of metadata, let's describe how the metadata standards can be used to describe the usage of CDISC throughout a study.

STARTING WITH ACQUIRING DATA

CDISC provides guidance with the development of Clinical Data Acquisition Standards, CDASH. The use of CDASH provides a series of standards for how data can be described for external systems, such as electronic data capture, EDC, systems can collect data. Data is collected by questions organized within a form. Each form has a label as well as specific attributes on how it can be presented. Each variable also has a label describing what is being asked of the investigator in order to fill that value. For some questions there are specific lists available which control what can be entered.

Based on what we have described, the form becomes the element set while each question becomes an element. For the form, we can add attributes to describe the name on the form, such as the Demographics or Study Drug Exposure form. The questions can also have labels such as "Age of Subject at Randomization" or "Number of pills taken". By doing this we manage how the content of the data is collected. This is all shown in Figure 2.

RELATING TO SDTM

Now that we have described what we are collecting, we can also build the relationships to the elements needed to build the Study Data Tabulation Model. In this way, elements are just elements as we have also shown in Figure 2.

The key to managing this is not just documenting the element sets and elements. The relationships also have to take on a meaning. Each relationship must also have attributes which describe the relationship

between two elements. These attributes can be a description on how to transform the data from one element to the next. In this way you can identify the source of a piece of data, the target for the data, and what needs to be done to the data to get it from one point to the next.

Translating that to CDASH, we can collect the data on the different forms and maintain the context of the data collected. By adding the relationships we can accurately map the data collected to the SDTM data structures and include the derivation on how to move the data.

ADDING IN ANALYSIS DATA, IN REVERSE

As we have now explained, you should be getting an appreciation for the element sets, elements, and relationships and how this works. That means that the Analysis Data Mode, ADaM, can also be represented using the same metadata constructs. In this case, ADSL becomes the element set while the variables become the elements. The ADaM Basic Data Structure can also be represented using the same elements with the inclusion of Value Level metadata, as described in the metadata definition.

The simple approach is relating the SDTM pieces to the ADSL data structure, at least for those items that are common such as subject, age, race, and treatment to name but a few. The greater challenge is in elements that are pure derivations, such as the population flags. For these, each element can have a derivation attribute assigned which shows how the population can be assigned. Because the population flag will generally be based on SDTM variables for the derivation, the relationships now become the SDTM variables used in the derivation. At this point, we are now beginning to develop the chain of control from the derivation to the source.

For value metadata within the metadata architecture, the challenge is much more complicated due to the way ADaM develops the data structure. The value metadata is based on how the data itself will be used. For this purpose, the Analysis Results Metadata (ARM) should be built by creating a table as the element set and each computation within a table as the element. The attribute for the element becomes a label describing any detail as well as the calculation being performed. Once that is completed, then the ADaM elements are created based on the needs of the calculation elements. Once the ADaM elements are created, they can be grouped into element sets for the purpose of defining the ADaM datasets.

Remember that the key is in the relationships. By creating metadata which shows the ARM elements and the ADaM elements needed, the next step is to define the relationships between them and include any derivations needed to convert the data from ADaM to ARM (such as formatting or such.) Since ADaM is dependent on SDTM, the ADaM elements will also need relationships created to the SDTM source.

Through all of this, we can begin to see the traceability of data from not just data collection into SDTM but also from Table calculations back to ADaM and finally the SDTM and CDASH elements.

KEEPING TRACK OF EVERYTHING

As you can tell, there are a lot of pieces to the CDASH, SDTM, ADaM, and ARM element sets, elements, value domains, values, and relationships. Doing this via written specifications, or an excel spreadsheet, can create considerable effort for a team. Tracking changes that occur during a normal specification process will also overwhelm a small department of programmers and specification writers.

To aid in making this process manageable, one should look to the use of a metadata registry (MDR). The purpose of an MDR is to capture the details of the elements and relationships in a way that can be traced. Building automation into the relationships also helps so that the defining of the relationship only has to be done on one of the elements but is then inherited by the system through automation. Other elements of an MDR include the creation of global standards and governance which can allow short cuts in the creation of study specifications from beginning to end.

Once the MDR is in place, the next stage is to build the automation in creating the submission datasets. Since the specifications are now in a machine readable format, the process for reading and automation is simply the application of data driven programming, a concept that we are all familiar with.

CONCLUSION

The complexity of creating SDTM and ADaM datasets is not going to be simpler. The creation of more complex trial designs and new data sources will continue to push CDISC programmers in how the data can be mapped and traced from beginning to end. At the same time, the push in industry is to find more creative ways to reduce the amount of effort used to create submission ready packages.

The use of metadata, with a metadata repository, will allow teams to capture more of the details in a way that will allow for the automation of reporting. While it may never be feasible to fully automate the creation and delivery of analytics (the push button report), we should as an industry push more for the automation of basic reporting concepts. By doing so we allow ourselves more time to focus on better analytics for understanding the clinical trial data we have received.

REFERENCES

Information Technology – Metadata Registries (MDR), ISO/IEC 11179 prepared by the Technical Committee ISO/IEC JTC 1, Information Technology Subcommittee SC32, Data Management and Interchange.

ACKNOWLEDGMENTS

The authors would like to recognize Barry Nelson, Chief Standards Officer, Nurocor, Inc.

CONTACT INFORMATION

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

Paul Slagle
Director, Global Data Standards, Clinical Data Services
Syneos Health
Paul.slagle@syneoshealth.com / pkslagle@gmail.com

Eric Larson
Senior Engineer, Global Data Standards, Clinical Data Services
Syneos Health
Eric.Larson@syneoshealth.com / larsonej@me.com