PharmaSUG 2022 - Paper RW-134

Redefining Industry Standards: Making CDASH and SDTM Work Together from the Database Level

Chad Fewell, Clinipace and Jesse Beck, Bristol-Myers Squibb

ABSTRACT

In today's industry, there appears to be a disconnect between the purpose of designing a database and how the database can be utilized to represent a client's needs better and more efficiently. The database is not just a tool for capturing data. It is also a tool that can be standardized to provide SDTMs almost instantaneously. The industry has moved towards CDASH being the standard for database design however there appears to be a fundamental misunderstanding of "Why" CDASH is needing to be used. This has led to a lot of the efficiencies that CDASH can provide to never come to fruition for SDTM creation. Subsequently, this leads into the need for a redefinition of the industry norm when creating a database. This inefficiency in CDASH primarily comes from a lack of understanding of how both CDASH and SDTMs work together in tandem. When a database is designed in RAVE, TrialMaster, or any other electronic database system there needs to be a strong SDTM influence into variable design as well. There is a flexibility in CDASH variable naming that just cannot be duplicated within SDTMs as well as the ability to preemptively check for data entry errors typical checks do not look for. The flexibility of naming for CDASH variables can be leveraged into a standard process which could ultimately allow for SDTMs to be written and Pinnacle 21 checks designed at the CDASH level all while abiding by CDASH and SDTM standards and naming rules. However, this can only be done if the database design is created by an individual or team of colleagues that are experts in both CDASH and SDTM.

INTRODUCTION

Data is used and stored across databases all types of ways. This paper is to help users understand how the variables within CDASH help create the SDTMs used in analysis and to help standardize a better process of creating and handling of a database. The current industry standard for designing a database is a clinical database programmer with input primarily from data management creates a database following the CDASH standards. This form of database creation is done with little input from the individuals that will have to handle the adaptation from CDASH to SDTM. One could argue it is imperative that the individuals converting data from CDASH to SDTM have influence within the CDASH variable creation as there are a plethora of tools programmers can use to efficiently transition CDASH into SDTMs. CDSIC has done a wonderful job of providing resources of how CDASH transitions into SDTMs. The thing lacking from these resources is the nuance into how to handle data flexibility within a database while also not jeopardizing CDASH to SDTM transition. What this paper will strive to do is describe a standard process allowing for individuals impacted by database design to have input into database creation ultimately leading to a quick efficient database that converts raw data from CDASH into SDTM.

COMPARING NOW VERSUS FUTURE

CURRENT SYSTEM

In today's industry, there are a lot of silos for different departments. While this sometimes happens to make people within those silos more efficient as they are more focused on a specific task, it limits the individual from being able to see the entire picture. Much like an engineer that spends their entire life designing a specific part for a plane. They may understand how their part operates and fits within the plane. However, they may not understand how it helps the plane fly.

Currently, databases are designed efficiently and quickly towards client needs. This is in no way a knock on how databases are currently built. It is to inform readers of possible ways to make database creation more efficient for all users involved. Databases tend to be designed with narrow understanding of how the data entered into the database will be utilized by SDTMs. There is a mentality of as long as CDASH is followed it will easily be converted into SDTMs. This can lead to an under utilization of the standards that CDISC has put into place for CDASH and SDTM. Database programmers tend to have extremely strong

background in Data Management and with the electronic database they are working on. This leads to databases being created quickly, with reliable edit checks, and reliable CRF design. Sounds great, right? Where is this missing link one may ask their self?

FUTURE SYSTEM

It is exciting to be able to build on a system. There is no leaving the current system in the past. However, there is a massive opportunity to grow the system and take the next step forward. That next step will be allowing individuals with SDTM expertise in on database creation. If SDTM experts could transcribe how the data entered onto the CRF will be transformed into SDTM there could be possible CRF design updates, or even remapping of CDASH variables. In addition, the SDTM insight could standardize how CDASH variables are identified and named. This could allow for SDTM programmers to insert global macros, controlled terminology, or better understand the raw data they are viewing without consulting data management.

Moreover, if the SDTM variables could be mapped all the way back to data entry this could allow for the development of Pinnacle 21 (P21) checks at the database level. This means a lot cleaner initial P21 reports and data issues that are not typically caught until SDTM generation. The future may seem like it is leading to complex database design, but it is just building that next step of efficiency on top of an already strong foundation of database design.

HOW TO IMPLEMENT THE FUTURE

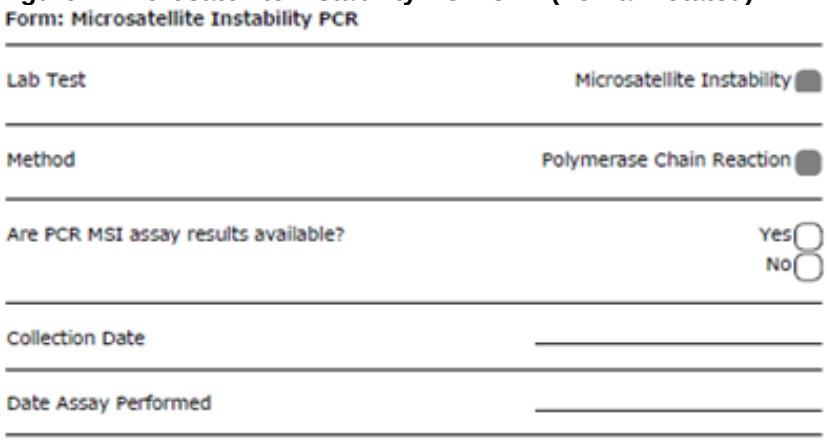
STEP ONE – CREATE INITIAL DATABASE

This process will follow the same initial process for database creation the current system uses. However, CDASH variables will not be fully assigned. This is due to the numerous reasons. One stems from there being numerous ways to implement the SDTM Implementation Guidelines (SDTMIG.) Another is due to allowable SDTM variables that may not frequently be used in databases. In addition, depending on therapeutic area or protocol there is a lot of grey area that leaves SDTMs up for interpretation. Currently there is not much, if any, process on a database level to handle these interpretations, which leads to our next step.

STEP TWO – APPROVAL OF ANNOTATIONS

This is really where the future system starts to shift away from the current system. This step would serve as a chance for the client and SDTM experts to begin applying their interpretations to grey areas. For example, lets look at Figure 1 below.

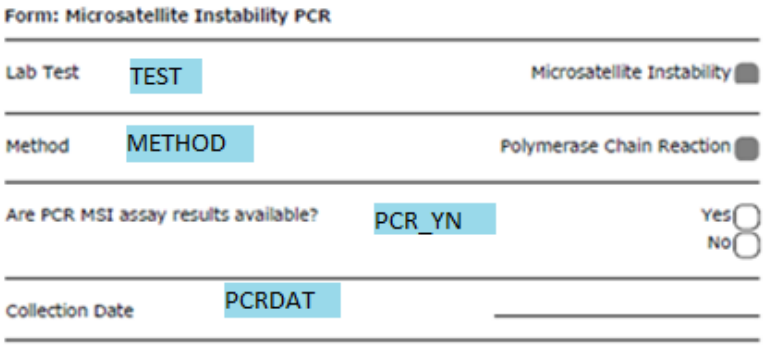
Figure 1. Microsatellite Instability PCR CRF (non-annotated)



This is a Microsatellite Instability PCR CRF. This CRF could be mapped to multiple SDTMs depending on interpretation. In addition, sometimes clients prefer to see data within specific domains they may or may

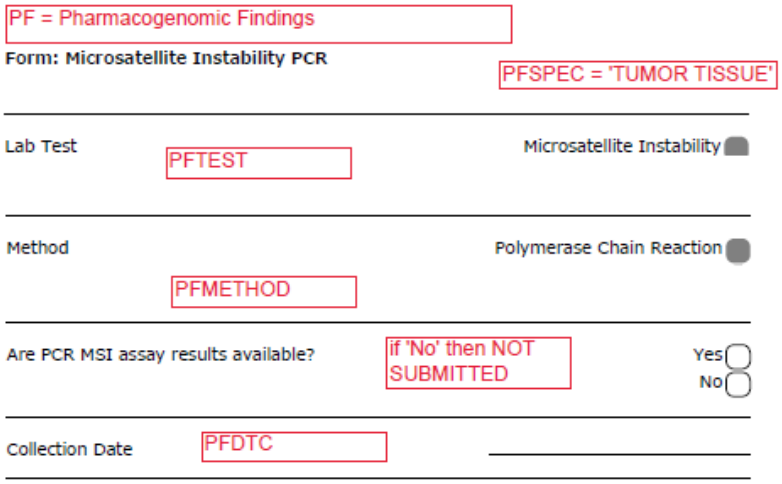
not follow CDISC recommendations. There could be numerous reasons for the client's preference however the CRF mapping itself will always be left up for interpretation. This typically means that when the CRF is designed the CDASH variables could be very generic in nature, see Figure 2.

Figure 2. Microsatellite Instability PCR CRF (annotated CDASH only)



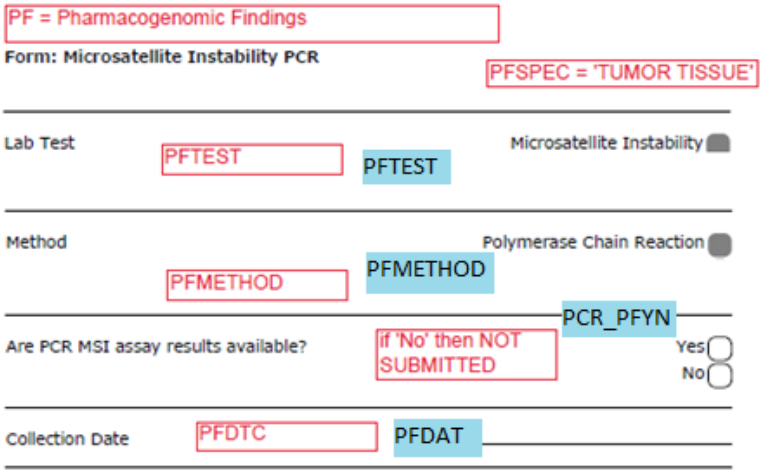
Different interpretations could have this CRF mapped to Microbiology (MB) or in this case the client wanted the CRF to be mapped to the Pharmacogenomic Findings (PF) domain, see Figure 3. This difference in mapping drastically alters the expected CDASH variables, and with the current process for database creation this flexibility cannot be addressed.

Figure 3. Microsatellite Instability PCR CRF (annotated SDTM only)



Now that there are SDTM annotations the database programmer can now revisit the CDASH variables on the database level and name them based off the interpretation of the SDTMs. This leads to SDTM specific CDASH variables that will drastically help when programming SDTMs. In addition, it will allow for the development of tools that can extract and convert the CDASH variable into their SDTM convention with little oversight. Figure 4 will illustrate what the expected CDASH variables would be named in blue.

Figure 4. Microsatellite Instability PCR CRF (annotated SDTM and CDASH)



This is a very simple example, but it helps illustrate how SDTM input on a database level can drastically change how the CDASH database variables are created and named. It also shows how CDASH variables are at times reliant on SDTM interpretation which the current process cannot account for.

STEP THREE – CREATE ADDITIONAL CHECKS AND VARIABLES

Now that the mapping between CDASH and SDTM has been illustrated the next step would be to adapt additional edit checks to account for more complex checks. Since the SDTM variable has been identified there could be checks that help SDTMs out with P21 reports. These checks do not have to be complex in nature. They could be as simple as making sure the CRF data is compliant with the SDTM variable expectations, or additional checks looking for data entry errors.

Not only can one expand on edit checks ensuring the integrity of the data being entered, one can now preemptively add in hidden values that are not seen on the CRF. These hidden values are CDASH and SDTM variables that are expected in CDISC standards. A prime example of one of these variables would be a PRPRESP, Procedure Prespecified, variable. Depending on the CRF this can be auto populated based on data entry. This would then move the derivation from the SDTM programming into the database. This ultimately helps ensure derivation accuracy and remove possible human error within the SDTM programming, which once again adds to data integrity.

ISSUES/SOLUTIONS FOR THIS APPROACH

ISSUES

One of the first issues with this approach is that it is experience dependent. One would need a team of not only experts, but also good communicators. Unless a single individual is an expert in all fields involved one would be looking at a team of individuals working together to implement this process. Sometimes more voices can lead to delays in implementation and disagreements on paths forward. Until individuals get experience with this process there could be significant delays in applying the process.

Moreover, the first few instances of implementation will result in a significant learning curve. There will be an entirely new area of expertise being addressed on the database level. This application of SDTM input would come with some process headaches. If there is a team working on the database, the SDTM expert will have to understand what capabilities a database programmer can and cannot do with the electronic database environment they are working within. In addition, the clinical programmer will have to understand why a CDASH variable will need to be adapted based on the SDTM mapping.

These issues will initially lead to time intensive database design. Clients already expect their database to be up and running as soon as possible. Adding complexity into the database is only going to acerbate those timelines. How does one handle the additional time that it will take to implement more complexity into the database?

Additionally, this approach can be limited by the database environment being used. Ideally, one would be using a database that can handle CDASH naming expectations. However, depending on the interface being used sometimes there could be instances that the database simply cannot handle the application of CDASH/SDTM naming conventions. The database may not be able to handle some of the P21 checks one could implement as well which would be something to take note of.

SOLUTIONS

Experience is not a bad thing. If one was to approach a client with this nuanced plan and expectations they have for the database, the client may feel more inclined to take this approach. Ensuring the integrity of the database is a worthy investment of time and resources. Reputation of a database can have a major impact on the analysis for the study. Ensuring the integrity of data on the database level instead of seeing issues on the SDTM level can also save considerable amounts of time. So, there is a time reward on the backend for taking more time on the front end of database design. This time saved would be seen at the SDTM/ADaM programming level.

Innovation always comes at a price. There will be multiple iterations, gradual adaptations, and success from failures. That is the price paid for innovation. Initially, I would expect this process to be pretty time consuming. However, as individuals become more exposed to the process, it becomes more fine-tuned, and individuals understand how the process works it will become standardized. This would lead to drastic reduction in timelines long term, with the hopes of reducing them to what are expected for today's standards. One thing that should be pointed out with this process is there is nothing new being implemented. It is moving more of the post database live processes, SDTM processes, into the database creation level. While the timeline will appear longer on the startup phase, the overall timelines of the clients being able to see their data remains the same.

In addition, using database creation phase to add these processes would be ideal as typically when a database "goes live" there is little to no data available. It is imperative that this early phase of studies should take the time of minimal data to set up the best system that gives the data its highest integrity.

The other thing solved by this approach is flexibility for clients. Yes, one may be limited by the flexibility within the database. However, this approach allows for maximum flexibility with the client. As illustrated in the figures above, any grey areas that are left for interpretation now have the flexibility to be accounted for at the database level instead of the SDTM level. There is a drastic increase in the flexibility of CDASH variables making them more compliant with CDISC expectations. This would result in less P21 findings and more efficient SDTM programming.

CONCLUSION

In sum, this article is to help provide insight on how one can build upon the current database creation standard. Each additional step taken on the front end of database creation can drastically improve backend results and reduce timelines to provide clients with reliable data. Once this process is implemented and experience is gained hopefully database timelines would stay around today's industry expectations. In addition, using this process would create of a position of industry expertise. The position would groom individuals to have both CDASH and SDTM expertise. This level of expertise would help reestablish the industry norm and help reset the industry expectations for databases created in the future. In addition, this approach will increase opportunities to utilize programmatic approaches for data visualization tools with this implementation. The industry is moving towards the expectation of providing clients with data visual tools as soon as the database goes live. This standard process and implementation would provide a stable foundation from the database level which would result in a drastic increase in opportunities to provide clients just that.

CONTACT INFORMATION

Chad Fewell
Principal Statistical Programmer, Clinipace
(615)887-3001
cfewell@clinipace.com