Paper TT07

The Elusive Goal of Automation in SDTM Mapping: A CRO perspective

Sridhar Vijendra, Ephicacy Lifescience Analytics, Bangalore, India
Tyagrajan Swaminathan, Ephicacy Lifescience Analytics, Bangalore, India

ABSTRACT

Much has been proposed and experimented with automating SDTM mapping to convert raw data (legacy/CDASH format) to CDISC SDTM standards with no or minimal human intervention. This is no easy task given the complexity and variety of studies, constantly evolving standards, and the struggle to keep up with demanding timelines. Stakes are higher in a CRO (Contract Research Organization) environment where additional variables sneak in to make SDTM automation a near impossible feat. One of the ways to automate SDTM mapping is to standardize CRF pages using a metadata repository (MDR) that allows usage of standardized mapping programs. The other option is to use machine learning and natural language processing, but this is yet to see effective implementation in practice. In this presentation, we look at SDTM mapping needs for our client to find an option that minimizes human intervention to speed up turnaround with no compromise on quality.

INTRODUCTION

SDTM Mapping is a key task in the line of activities leading to submission ready artefacts. A number of solutions have been proposed for easing the task of mapping the raw data captured by EDC (Electronic Data Capture) systems to SDTM compliant datasets. Transforming the raw data into SDTM-compliant datasets sounds like a fairly simple process but is quite a tedious task due to several factors. These factors include but are not limited to - ever-evolving standards, difference in EDC systems within/across organizations, study design variations, and compliance to company-specific standards. High level of CDASH compliance of the EDC is not enough to ensure automation. Often, each new SDTM mapping scenario within an organization is unique enough to warrant reinvention of the wheel to optimize the process of mapping. Automation for lightning fast, error-free SDTM mapping is still far from reality. Expensive tools for automation are available and proprietary tools are usually developed in-house in many organizations. However, use of automation tools for SDTM mapping within statistical programming teams is not yet widespread. In addition, SDTM mapping automation is generally incomplete without some amount of manual intervention. This ensures accurate mapping in various study design scenarios.

THE METADATA REPOSITORY

The concept of an MDR (Meta Data Repository) has picked up rapidly in the past decade and most organizations handling clinical data have by now adopted some form of an MDR. In short, an MDR is essentially a library of mappings from the Case Report Forms (CRFs) to the SDTMs and beyond. One of the highlights of an MDR is the benefit of data traceability it provides. It also allows change propagation of metadata from the source to the target. MDRs are meant to be EDC-independent, which means mappings can exist for different EDCs simultaneously.

Having an MDR allows the creation of a case book of CRFs for a study from the available CRF library. This can be done for any study design scenario. Once the case book of CRFs is ready, the mapping of each of the fields in the CRFs are already available in the MDR. Apart from the vast data management potential of an MDR, MDR-based SDTM mapping has been discussed widely [1, 2, 3] and is well established. If macros are written once to execute these mappings, they can be used repeatedly. This automated mapping will be interrupted for a new study design scenario that might bring in new CRF fields or a new CRF. At this stage, a macro module may be added. This process is repeated over many new scenarios.

In many instances, there could be a layer or more between the raw data extracted from the database and the SDTM layer. These intermediate layer(s) are used for the purpose of data review. This is because data review is generally easier with a non-SDTM kind of structure. These intermediate layers can also be included in an MDR. In that case, SDTM mapping starts with the intermediate layer as the source.

CRF annotation and other associated tasks in SDTM mapping can also be accomplished easily with an MDR. Over time, the MDR can support significant automation and bring manual intervention down to making study specific changes and performing the review of SDTMs. It is apparent that MDR-based systems require a significant amount of work at the outset. This includes configuration, creation of a robust code base, and training of teams to

work with the MDR. The literature is weak on the difference in metrics between a manual mapping activity and an MDR based automated mapping over a long run. In any case, MDR-based mapping allows for standard consistent SDTM outputs to be produced, with some amount of manual intervention thrown in.

RULE-BASED SYSTEMS FOR AUTOMATION

Rule-based systems have been around for a long time. Simple rule-based systems, also known as expert systems, give the impression of mimicking human intelligence. In reality, such systems follow a set of if-then-else conditions to produce a result for a scenario that has been seen before. For any new scenario, such a system would simply produce unintelligible or no output. Obviously, MDR-based mapping is a rule-based system tied very closely to the standards we use. A more complex rule-based system can be built that contains the MDR. However, a rule-based system without MDR works best for legacy or non-standard input data. In any case, such rule-based systems need constant updates to deal with new inputs that the system has not seen. There is no scope for self-learning in such systems, which is where artificial intelligence or machine learning comes into the picture.

MACHINE LEARNING FOR AUTOMATION

Several papers and presentations [4, 5] have brought up the use of a combination of machine learning and/or natural language processing for automation of SDTM mapping. Self-learning systems are a step ahead of rule-based systems. Such systems can adapt to new inputs and attempt to produce a desired output. However, such systems first need to be trained with various kind of inputs to allow for them to learn. A significant amount of research and development would have to be done in-house to create a validated system that would be capable of churning out highly accurate SDTMs at a quick pace. This is assuming the possibility that machine learning and related systems can provide the near-perfect solution we are looking for. It is obvious that an MDR would be part of such a system by default since an MDR satisfies the basic need of standardization. The automation will be taken care by the rest of the system intelligence.

THE CRO PERSEPECTIVE

The sponsor-CRO collaboration is a well-established trend of the pharmaceutical industry. CROs ease the burden of sponsors for several tasks and one such popularly outsourced task is SDTM mapping. From a CRO point of view, manual SDTM mapping using a skilled team of SDTM programmers is straightforward. However, manual mapping of SDTMs is tedious, particularly due to the data intensive nature of the task. In some cases, the contract between sponsor and CRO allows for the mapping to be done in-house within the CRO systems. This gives more room for the CRO to establish and use a robust library of macros or any proprietary validated system to generate and deliver the artefacts. For CROs that work on sponsor systems, typically in an FSP (Functional Service Provider) contract, it is harder for the CRO to come up with a one-size-fits-all solution to automate SDTM mapping. The only way forward would be to optimize certain processes and tweak specific parameters in the mapping processes to ensure quicker turnaround.

THE SPECIFIC MAPPING PROBLEM

The specific problem we were faced with had its unique challenges, just like every other SDTM mapping scenario. The client was looking at a quick turnaround of SDTMs mapped manually. The starting point of the mapping process was with SDTM specifications that were provided along with the annotated CRF for the study. The raw data available to us was in the form of Inform® raw views extracted from the Data Management Workbench (DMW®). In terms of data layers, as shown in Figure 1, there was one intermediate layer between the raw Inform® views and the SDTMs. This intermediate layer, which will be referred to as the ***Layer2*** data, was generated by an automatic set and merge program. The program took a best guess with key variables between the raw views for a particular domain and created a single Layer2 dataset for that domain. For instance, if disposition data had 6 views in the raw data layer, a single Layer2 DS dataset would be created by the program which could be used as the source for the SDTM program. The specifications were tailored based on the Layer2 data as source in most cases. In some cases, the automatic merge created an inaccurate Layer2 data for a domain and in that case, the raw data was used as source. This also meant that the cross-checking of the automatic merge was one of the subtasks in the mapping process.

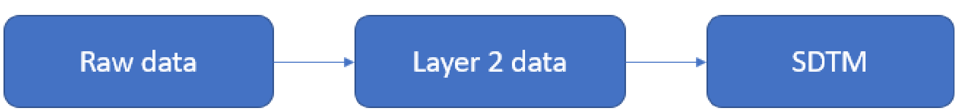

Figure 1: The intermediate layer between raw views and SDTMs

Figure 2 illustrates the various subtasks that were part of the process. Each subtask had its significance in delivering accurately mapped SDTMs. The task before programming involves the cross-checking of the raw data with the Layer2 data.

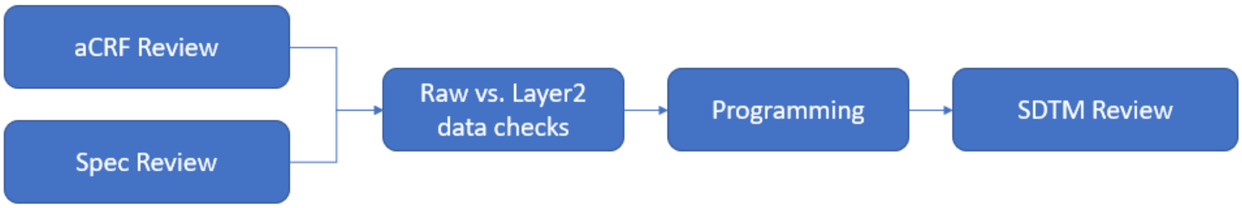

Figure 2: The sequential tasks in the specific SDTM mapping scenario

MACRO LIBRARIES

As with many client-CRO partnerships, the client provided us access to their macro libraries. Client-supplied macro libraries can be a blessing, but they have their disadvantages too. The positive side of the macro libraries is that it obviates the need for the CRO to come up with its own. It also has gone through significant testing already and allows easier compliance to client-specific standards and standard operating procedures (SOPs). The limitations of such macro libraries for the CRO programming team include working with poorly documented macros and the associated learning curve. Macro libraries that have evolved over many years and through many standards have been tuned well enough for churning out desired results with the right inputs, but teams usually take a while to utilize their full potential. Although we had the tacit approval to write our own macro library, the demand of quick turnaround left the team very little room to start writing macros from scratch. We were also unable to re-purpose our in-house SDTM macros for the specific mapping scenario due to the same reason.

In any case, there was a significant amount of metadata-driven programming already coded into the macro libraries. Metadata-driven programming is illustrated in Figure 3. The Layer2 metadata was used as the input for these metadata-driven macros. In conjunction with metadata mapping tables, up to 75% of the variables in every domain were directly mapped to their SDTM counterparts. The remaining 25% needed to be checked and mapped manually. Whereas the amount of "automation" in this mapping was appreciated, it was observed that even the 75% of mapping had to be re-visited for every new study to ensure it was producing the desired result.

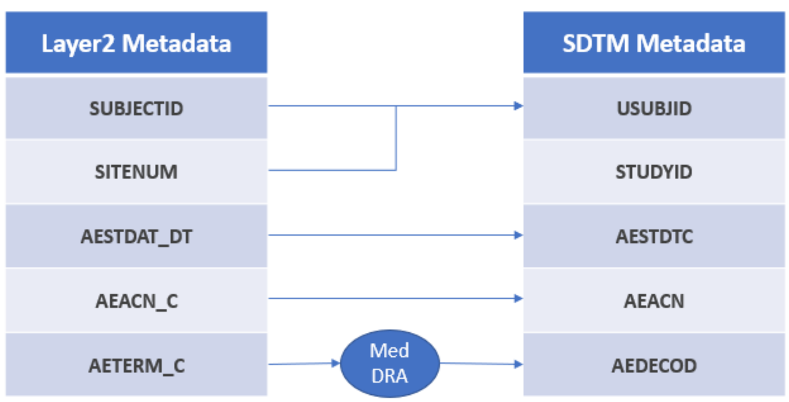

Figure 3: Metadata-driven mapping of SDTM

TASK-WISE EFFORT BREAKUP

After a good number of studies, it was possible to identify the amount of time spent on each subtask. In terms of percentage of time spent in each of the subtasks shown in Figure 2, the actual programming was a fraction of the time spent on the mapping process. Figure 4 below shows the breakup in terms of percentage of time spent on each domain. The specification review, aCRF review and verification of the data layer took up a big chunk of the time required for mapping.

Figure 4: Percentage of time spent in mapping a single domain

SCOPE FOR AUTOMATION

In terms of automating or in the least, optimizing the mapping process, we came up with a few options:

- The first option was of course to consider overhauling the entire process to automate it. A few brainstorming sessions later, it was apparent that the scope of automation for this task was limited due to several reasons:
  - The quick turnaround time that was expected required us to dedicate our entire team to delivering the studies allowing little scope for research & development at the outset
  - The sponsor had plans to overhaul the process. The sponsor was planning on changing the standards of the Layer2 data and creating another macro library for the same. This meant that we should consider coming up with a generic solution that addressed various scenarios rather than addressing a specific scenario that was going to be obsolete soon

- The second option was to consider applying MDR-based mapping directly from the raw Inform views to SDTMs. This was not desirable because this would skip the crucial data review layer. Creating a separate track of data from Inform to SDTM while still retaining the review layer would affect traceability. Hence, this option was discarded.

- The leftover and not-so-exciting option was to consider optimizing certain subtasks in the mapping process. It was clear that none of the subtasks could be skipped. The spec and aCRF review and the cross-checking of the raw with the layer2 data were 2 tasks where little could be done. What was left was the programming and review subtasks. With the metadata-driven mapping already helping us reduce programming time, there were few options to optimize programming further.

INCREMENTAL PROGRAMMING

One of the ways we decided to optimize the programming step was to reduce the time needed to re-program domains that were already programmed in an earlier study. Over the course of mapping 8-10 studies, we realized that programmers were perusing the specifications for each study to program the domains. While this was needed for the purpose of doing a thorough review, it was not needed for doing the programming or for the review process. This was because there was a lot of code that could be re-used from study to study in spite of differences in study design. The key difference was in the number of SDTM variables in each domain and in the business rule for many of their variables. For instance, from one study to another, the LB domain could have only 2 variables more and only 1 of the remaining variables would have a difference in the business rule from one study to another. For a programmer who has just looked at the specs and the aCRF, this difference with the other study would not be apparent and when the programmer starts programming, they would start from the first line of the specs and go through it variable by variable. This can be time-consuming and wasteful. A simple difference output between the new study specification and a reference study specification would be able to ease the programming effort. This effort could be directed towards a thorough review.

Instead of looking through the specifications completely and programming each domain fresh using the macro library, it would save a lot of time and effort if a compare output were available between the new study specifications and a reference study that was programmed earlier. The highlighted difference provided by the compare would make it easy for the programmer to make the incremental change and then move on to the SDTM review. This is illustrated in Figure 5.

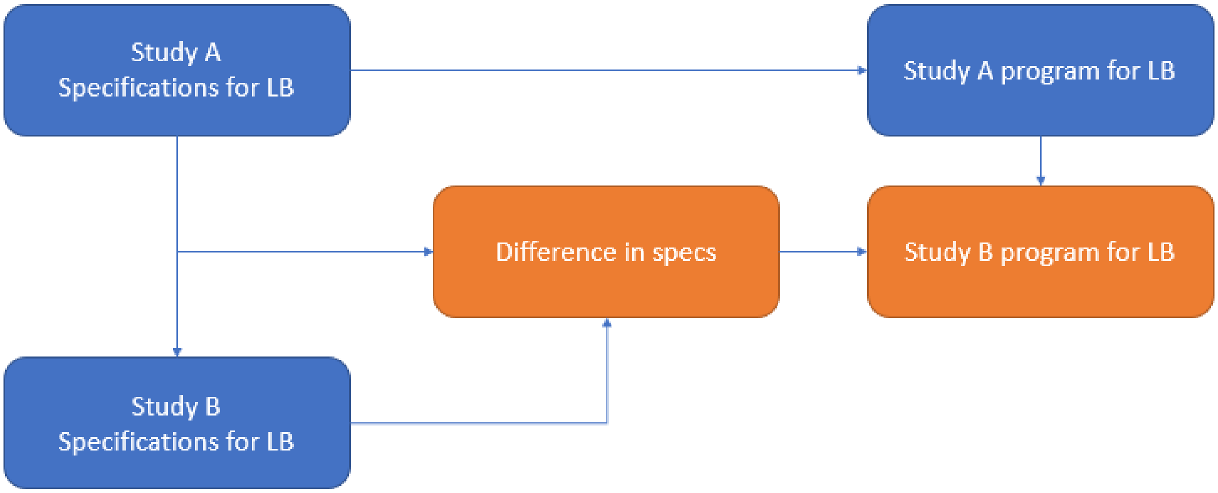

Figure 5: Incremental changes to code

RESULTS AND NEXT STEPS

The incremental programming helped reduce the programming time and minimize validation comments in subsequent studies. It also helped maintain consistent code quality. In the studies in which it was implemented, it reduced the programming time by about 50% and reduced the overall completion time of the study by a margin. It can be argued that such incremental programming can take the programmer's focus away from the data. However, with a data review, spec review and aCRF review subtask already being done, putting the program together quickly can allow increased time for review and better focus on data.

SPECIFIC VS. GENERIC SOLUTIONS

While incremental programming can marginally optimize the programming, what really is needed is a generic solution for automating or almost-automating SDTM mapping. This solution would need to be diverse enough to be applied for mapping problems presented to a CRO by various sponsors. This would mean that such a solution should be readily movable into the sponsor system to enhance mapping efficiency and should be adaptable to different kinds of input data standards and systems.

The promise of learning systems is very attractive and provides a lot of scope for flexibility and improvement of the SDTM mapping process. Our current exploration in learning systems used in conjunction with an MDR hopes to take us closer to actual automation of SDTM mapping. Hopefully the results will be far more impressive than this simple optimization process.

CONCLUSION

SDTM mapping is one of the key tasks on the way to a successful submission. A seemingly simple task, SDTM mapping is tedious, closely tied to various standards, and fraught with challenges in the details. Every SDTM mapping scenario can bring up a new set of challenges, especially for a CRO providing mapping services for one or more sponsors. In such cases, an automation solution for one scenario will not apply to another and there is a significant amount of re-work for every project involving a group of studies. For CROs working on sponsor systems, the challenges are greater since there is only so much systemic change the CRO can make in the sponsor systems to provide a feasible solution.

We were presented with a specific SDTM mapping scenario by our sponsor to turn around many studies in a short time. We were at odds with identifying a suitable strategy for reducing turnaround time while maintaining the same quality that manual mapping can provide. We considered systemic solutions such as MDR-based mapping and toyed with the idea of exploring advanced techniques such as machine learning for solving this problem. However, given the time constraints involved, optimizing the current process seemed to be the way to go. By breaking up the task of mapping into subtasks and identifying the subtasks which could be optimized, it was possible for us to cut mapping time for each domain by a small margin.

Excluding this specific scenario, it is apparent that solving this problem of automation can be attempted in various ways. These solutions could be solely based on an MDR, or a combination of an MDR and a feedback-based learning system. The possibilities are many but a near-perfect workable solution is still a dream.

REFERENCES

1. SDTM Automation using standard CRF pages, Paper PO21, PharmaSUG 2016.
2. End to End SDTM Automation: A Metadata Centric Approach, Paper SI01, PhUSE US Connect 2019.
3. Automating SDTM File Creation: Metadata Files Speeding the Process, Paper 181, SAS Global Forum 2010.
4. CDISC SDTM - An Automated Approach, Paper SI11, PHUSE EU Connect 2018
5. SDTM Mapping based on Natural Language Process and Machine Learning, www.cdisc.org

CONTACT INFORMATION

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

Sridhar Vijendra
Ephicacy Lifescience Analytics
Bangalore
India
sridhar.vijendra@ephicacy.com

Tyagrajan Swaminathan
Ephicacy Lifescience Analytics
Bangalore
India
tyagrajan.swaminathan@ephicacy.com
www.ephicacy.com