PhUSE US Connect 2019

Paper DH03

From source data to SDTM: enabling a semi-automated conversion

Pieter Blomme, SGS, Mechelen, Belgium

INTRODUCTION

At SGS, we have implemented SDTM as our internal data format. This enables us to streamline all of the downstream processes and tools, to clean the actual end product and to provide SDTM, SAS datasets and define.xml shortly after FPI.

Input data can be received in various formats depending on the origin of the data: different e-source systems, diverse (e)CRF systems and external providers with varying degrees of standards knowledge. The input data will therefore have to be converted to SDTM.

Fortunately, most data conversions largely follow the same processing flow:

1. Mapping the source fields to SDTM
2. Essential processing (USUBJID, visit assignment, date conversion,…)
3. Complex SDTM processing (defining subject visits, defining EPOCHs, original to standardized result conversion,…)

The majority of these steps can be automated, with the following adage: more standardization leads to more automatization, ultimately resulting in less trial-specific work.

At SGS, this realization has led to a split-up of the data conversion into 2 large parts: the pre-conversion, and the actual SDTM conversion. The pre-conversion makes the source fields available in a conversion table. The SDTM conversion then maps and processes the information from this table to SDTM.

PRE-CONVERSION

The pre-conversion ultimately makes source data fields available in a mapping table and consists of several sub-processes:

1. Data loading and formatting
2. General data pre-processing
3. Source system specific data pre-processing
4. Creating the conversion table

DATA LOADING AND FORMATTING

The first step of the pre-conversion is loading the source data in the database and making it available for the data conversion. In practice, we encounter 3 types of source files: SAS, database exports and text files. These all come with specific challenges:

- SAS and database exports can have diverse versions, which may not always be compatible.
- Text files can be delivered in a wide variety of formats: csv, ascii, tab separated, data fields enclosed by double quotes or not,….
- We also have to resolve character set incompatibilities, such as conversion from single byte to Unicode

It is crucial to have a validated system in place to load these different source files without data loss. Our data loading process imports all source exactly as is. This includes special characters, which need translation, and administrative fields, which may not be needed in the final SDTM. We always keep an exact copy of the source in our database. This ensures that we can always trace and compare the data from the source to the converted output.

Although this process is automated for all known source, we continually have to detect and process edge cases and/or new formats to ensure data integrity. There is no guarantee that source will continue to be delivered in the same format: version changes may impact both the format and the structure of the data.

GENERAL DATA PRE-PROCESSING

Next, the pre-conversion handles 2 pre-processing steps that are essential to enable a source-independent conversion.

- Map the subject IDs to the correct USUBJID format that the sponsor expects.

The name and location of the subject id differs per source system, and often it requires combining multiple tables. Most source systems assign an internal subject number to records, which is different from the sponsor-preferred

subject identifier. In those cases a mapping may be required. Since we do not want our core conversion to depend on the source system, this is handled by the pre-conversion. The core conversion never sees the internal identifier and does not need to know how the USUBJID was derived.

- Assign a visit to each record.

Each source system has some kind of visit identifier, which ultimately has to be mapped according to the visit map. Sometimes this identifier is one field, but for other systems it needs to be combined from multiple fields. For eSource, there can be even more complexities. Like the USUBJID, this is clearly a task that is best suited for the pre-conversion.

The only task left for the programmer is to take the generated visit identifier and map it to the correct SDTM visit and time point. This is done using a many-to-one mapping table. The SDTM identifiers are sponsor-specific, so manual programming is always required here.

SOURCE SYSTEM SPECIFIC PRE-PROCESSING

Each source pre-conversion also contains some convenient pre-processing to make the data easier to use.

One such step is date conversion. Date formats can be very diverse, even within the same (e)CRF/eSource system. It is essential that the dates are converted to ISO 8601 as soon as possible, because a lot of derivations happen on date fields in the later stages of the conversions.

The actual conversion of a date to ISO 8601 is not that difficult, because the date format should be entered consistently in a (e)CRF/eSource system. What is often more difficult is detecting which fields are dates. Some source systems have metadata where this information can be consulted, but in most cases, the only way to find out is by looking at the name and content of a field.

Sometimes a list of known date fields can be compiled. In other cases, the name of the field can be an indicator. Often the best solution is to try pattern matching: if the content matches the known date format most of the time, it will be a date field.

We also have to account for invalid dates: in those cases the original value should be returned. Otherwise it would be hard to pinpoint the exact data issue. Incorrect date conversions would also impact later derivations of visits and epochs, which we should avoid.

Because of this, date conversion is a part of the pre-conversion that requires regular maintenance.

As we use a specific source system more, we tend to build other convenient functionalities into the pre-conversion to simplify later conversion. For example, quite often data is entered on different forms that actually belong together. This is often the case with meal forms, or dosing events with different start and end times. If we can detect these forms, we try to join these tables in a new help table in the pre-conversion. That way, we can map the help table, where linked information is present in one record, making mapping considerably easier.

CREATING THE MAPPING TABLE

The next step is to make the data fields available for mapping. (e)CRF/eSource systems define forms for the data capture. Clinical data is entered in these forms, but the structure of the forms are of course very system-specific. Some systems use small forms, which often fit on one page. This simplifies the forms, but often means that linked information is present on multiple forms. Other source systems use very long forms containing a lot of information, which leads to its own specific mapping problems. eSource systems are a special case altogether, since these do not have conventional forms to be filled out and are often structured drastically different.

Since we might want to map different fields from a single form to multiple domains, or multiple forms to one record, and to be able to run a standardized process for all source systems, we generate a mapping table, in which each possible field (= variable) of each form (= table) becomes a separate record. Forms are linked together with a form reference identifier.

We try to retain as much information as possible in the mapping table, but there are some fields (and entire forms) that are filtered out of the pre-conversion. This is the case for purely administrative fields that are often not even present on the printouts of the CRF, such as labels, metadata information, reference numbers used only internally by the underlying source system. Since these fields are not on the annotated CRF, they will not have to be converted. It is always possible for the programmer to re-add some of these fields if necessary.

On top of the mapping table, an in-house mapping tool has been developed, in which the programmer can define what should happen with each field. The mapping tool tells the conversion to which domain(s) a field should go and what other fields it should be linked with to create a record. It is also where the programmer can define the first simple data conversions, such as

- Decoding a field
- Taking a substring
- Combining values from multiple fields into a single value

For more information about the mapping tool, please see PhUSE US Connect 2019 paper SI01 “End To End SDTM Automation: A Metadata Centric Approach”, by my colleague Roman Radelicki.

While there are many differences between study set-ups, a lot of forms will look similar for the same(CRF) system. Some standard annotated CRF forms do not undergo large changes between trials for the same compound or client. This allows us to re-use the majority of the mapping. Study-specific work is of course always needed, and it is important to carefully review the form setup and the annotations.
Finally, a large conversion table is generated by combining the mapping with the actual data. The conversion loops through the records in each table and applies the mapping. The conversion table defines:

- How the field should be converted (eg. substring, decode, …)
- Which domain it belongs to
- Whether it is a new record, or should be linked to an existing record

This is a process that runs every time a new source data load becomes available, which is usually every day. New data is loaded, the pre-conversion applies the mapping and generates the conversion table. Our standard conversion package is programmed to use this table, rather than the actual source tables.
Because of this, it is source system independent. Changes to the pre-conversion do not cause this main package to break, and for a new source system, a pre-conversion can be set up largely independent of the main conversion.

SDTM CONVERSION

Thanks to the pre-conversion and a standard mapping table, we are able to generate SDTM datasets very quickly. Our conversion package will:

- Pre-convert the latest source data
- Create a conversion table containing all fields that have a mapping defined.
- The mapping table defines which fields should trigger the creation of a record, e.g. a lab result, a physical exam test or a dosing event. These records are inserted in the right SDTM table, with several derived fields already correctly attached: USUBJID, STUDYID, VISIT, --TPT
- The conversion then checks the conversion table for all mapped fields linked to that record, eg. the --DTC value, --STAT, --REASND and so on. These fields are also correctly completed for that record

This results in semi-converted SDTM domains. Of course, there are many steps still needed to get to correct SDTM:

- Standardized field derivations
- - Epoch derivations
- - --DY derivations
- Completed DM study fields, SV and SE records for a subject
- Calculating baselines and sequences

Thanks to the pre-conversion, these steps are no longer source system specific, but dependent only on SDTM and client specifications. We no longer have to deal with a very diverse set of forms, fields and data formats. Instead, we have SDTM domains, which have a clearly defined and stable structure: we know the variables and their format (date, character, numeric) for each domain.

DOWNSTREAM PROCESSES

The process outlined above means that we have SDTM datasets available very early on, which has allowed us to develop 2 key tools for data validation and cleaning.
Our QC tool is used by the clinical data managers during the set-up phase of the conversion to check the quality of the conversion and to flag errors, by checking dummy subjects PDFs against the resulting SDTM datasets. Since it runs on SDTM datasets, a single tool can be used for all source systems.
During the study production phase, we use a tool called All2One for cleaning. This tool contains checks and listings that check the quality of the incoming data and flag any irregularities, so that they can be queried to the source (if bad data) or to the programmer (if there is a conversion issue). The tool uses the SDTM converted data, which makes the checks and listings much more versatile and reusable.

CONCLUSION

Transforming all data from its source format to SDTM right at the start of the workflow has several advantages. It avoids dealing with different data formats, allows cleaning on the actual end product and enables us to use downstream SDTM-based data management tools.
SDTM formatted data is created in 2 steps: a pre-conversion creates a uniform conversion table from diverse source systems. This facilitates a source independent SDTM conversion process.
Ultimately this results in much less trial-specific work, because a lot of programming can be re-used:

- The pre-conversion is source system specific, but independent of client specifications
- The mapping is both source system and client dependent, but a lot of work can be duplicated and adapted from similar trials

- The main conversion is SDTM and client dependent, but largely independent of the source system.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:
clinicalresearch@sgs.com
EUROPE: +32 15 27 32 45
AMERICAS: + 877 677 2667
WWW.SGS.COM/CRO