PharmaSUG 2017 – Paper IB07

Building a Fast Track for CDISC: Practical Ways to Support Consistent, Fast and Efficient SDTM Delivery

Steve Kirby, Chiltern; Mario Widel, Eli Lilly and Company;
Richard Addy, Chiltern

ABSTRACT

Standardized data are so useful that sponsors are now required to provide study data to the FDA (CBER and CDER) using the standards, formats, and terminologies specified in the Data Standards Catalog. In practice that means following CDISC SDTM for tabulations content. Good planning is required to make sure that SDTM data are ready as needed to support regulatory submission; better planning is needed to have SDTM data available as needed to support all internal and external data consumers.

We will share some effective strategies that we have used to provide the study data as collected to data consumers in SDTM format from first patient first visit through database lock. Our presentation will focus on three key areas: Planning to collect what we will submit with CDASH and SDTM-friendly protocols; preparing to consistently implement SDTM with metadata standards; and designing robust, reusable mapping code that is validated early and used often.

INTRODUCTION

Companies have always consistently worked to collect data as needed to fully support study objectives and endpoints. And while the most critical task is (and always will be) ensuring that protocols and eCRFs contain/collect all the information needed to objectively judge whether an investigational product is safe and effective, it is increasingly important to ensure that the information is easy to use and evaluate across studies and sponsors.

How can sponsors ensure that regulatory reviewers and other downstream consumers can easily use and evaluate their data? By building a process that makes data in standardized (CDISC) format available from the time of first collection through regulatory submission. It is easier to say what a process should accomplish than it is to build it. To help people build (or refine) their CDISC data processes we will share a few effective ways to streamline the path to SDTM.

Many of these approaches are based on the obvious: The shortest path to SDTM is to standardize collection forms with SDTM in mind; and standardized data are (well) standardized – take advantage of having stable, consistently formed inputs when designing mapping processes. Spending the effort to establish robust data processes will save time and money in the long run as code can be generated once and re-used as needed. While obviously sensible to support process refinements designed to better support CDISC, change is uncomfortable and people tend to avoid discomfort. So as you (and, yes, we are expecting each of you to do your part) work to streamline CDISC data processes, don't hesitate to do a bit of cheerleading.

What do we want? SDTM!

When do we want it? Now!

(Repeat as needed)

Ready? OK!

Read on for a few practical suggestions that can help turn a bit of enthusiasm into a faster (and easier) path to SDTM.

SDTM FRIENDLY PROTOCOL PROCESSES

Let's start at an early stage of the clinical trial planning process, protocol development. With the transition to practical details it is now time to move the theme from cheerleading to advertising. But don't lose your enthusiasm! What do we want? SDTM! When do we want it? Now! How can we do that? One good way is to build a process that ensures protocols play nice with SDTM. Doing that is not as easy as it may sound; and it is necessary to have CDISC experts closely involved with protocol design. Embrace the challenge, and move boldly forward!

We will share three ways that you (yes, you – we still expect all of you to do your part) can update processes related to protocol development to better support SDTM. It is good to do some minor adjustments to support controlled terminology; better to build in support for trial design domains and best to drive the protocol process across a submission with metadata that supports all the trial design domains. We know that you (and all the data consumers you support) deserve the best. The process suggestions below can help you get there. Interested? Read on to see how to get where you deserve to be.

GOOD PROTOCOLS FOLLOW CDISC CT

Have you ever seen a CRF pick list that just won't cooperate with CDISC CT? Hate that, right? The root cause may be in the protocol. Data collection is (you guessed it) based on the protocol. Plan to avoid downstream issues by making protocol language consistent with CDISC controlled terminology where applicable.

CDISC CT is potentially applicable to many protocol items. A couple representative areas that often should be better harmonized with CDISC CT are: Action taken in response to adverse event and reasons for discontinuation from treatment and/or study. At a minimum, study protocol review should specifically include consistency with CDISC CT, and protocol templates should be designed to be consistent with CDISC CT where applicable.

BETTER PROTOCOLS CLEARLY SUPPORT TRIAL DESIGN DOMAINS

Trial design model (TDM) domains database the protocol and summarize the study. The TDM domains are: Trial Arms (TA), Trial Disease Assessments (TD), Trial Elements (TE), Trial Inclusion/Exclusion (TI), Trial Summary (TS), and Trial Visits (TV). Unlike other SDTM content, TDM domains are primarily based on planned study conduct. TDM content can usefully support SDTM mapping for other domains and is an easily accessible reference for study conduct in general. All things considered (as suggested by the FDA) TDM domains are great to have around. So why is it common to avoid generating TDM domains until the last possible moment? One key reason is that protocols are typically not designed with TDM domains in mind, making what could be an easy process into a painful (and often delayed) translation exercise.

As is often the case in clinical research, keeping the end (in this case TDM domains) firmly in mind when designing protocols can help keep the focus on what is needed most; and as we will see in the next section, content needed to support TDM domains can be the starting point for protocol development as well. Below are a few representative examples of how protocols can (and should) support TDM domains. While all TDM domains deserve better protocol support (and can even support the protocol when done first) we are limiting discussion of this topic to Trial Summary (TS) and Trial Visits (TV) as those two domains nicely highlight the need to keep TDM domains in mind when writing protocols.

The Trial Summary (TS) domain gives an overview of the study. The number of parameters expected to be included has increased dramatically over time. The focus of TS is on information needed by FDA reviewers and there is a close relationship between the content in TS and in clinicaltrials.gov. Having the list of required and expected parameters in mind while developing the protocol greatly simplifies the effort involved in translating protocol prose into TS data.

Before we get to a few cases where protocols don't gracefully support TS, it is worth emphasizing that many TS parameters often are easy to determine. The protocol title alone will typically support many TS parameters. In figure 1 below, it is clear that TSVAL where TSPARM = TPHASE is "Phase I Trial"; TSVAL where TSPARM = TBLIND is "DOUBLE-BLIND"; TSVAL where TSPARM = TRT is "DRUG A" and (as a

separate record) DRUG B; TSVAL where TSPARM = HLTSUBJI is “Y”, etc. In fact the protocol title in coordination with the protocol synopsis often will gracefully provide the bulk of the values needed for TS.

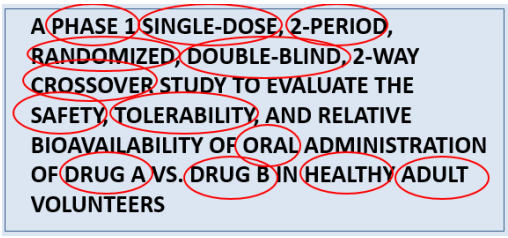


Figure 1. Protocol Title - TS Information Density

Even where the translation from protocol prose to TS values is easy, it is worth investigating how it can be made easier still. A few thoughts on that topic are coming up in the next section. And, as many of you know all too well, there are some TS parameters that are consistently challenging. For those cases, it may help to make a list of items that you would like to see supported in the protocol (or in an associated reference document). Then check to see that the information is available as needed during protocol review and suggest that the content be included in the protocol template.

Table 1 below shares basic protocol implementation notes for a few representative parameters. At the end of the day, it is rare that a required parameter is not supported by the protocol at all, but the available content often does not clearly match what is needed in TS. Focusing on how the information needs to live in TS (keeping controlled terminology references and related conventions in mind) when writing the protocol can help avoid translation issues and investigative cost. Someone writing the protocol has the information needed for TS; they just are not documenting it as needed.

PARAMCD	PARAM	Protocol Implementation Notes
TRT	Investigational Therapy or Treatment	TRT (and CURTRT, COMPTRT) are subject to the UNII (FDA Unique Ingredient Identifier) codelist. Good to ensure that UNII references are provided or supported for each treatment ingredient for each treatment.
TDIGRP	Diagnosis Group	When applicable, TDIGRP (and INDIC) are subject to the SNOMED codelist. Good to ensure that SNOMED references are provided or supported.
OBJPRIM	Trial Primary Objective	Best to avoid compound objectives and to clearly designate whether objectives are Primary or Secondary.
PCLAS	Pharmacological Class of Invest. Therapy	PCLAS is subject to the NDF-RT (The Veterans Administration’s National Drug File – Reference Terminology). Good to ensure that NDF-RT references are provided or supported if known and otherwise documented as not available.

Table 1. A few Trial Summary Parameters with Protocol Implementation Notes

The Trial Visits (TV) domain summarizes scheduled timing and contains one record per planned visit (“clinical encounter”). Visits are defined in the protocol. Typically the best source of information is the schedule of assessments. Having a clear and easily accessible set of visits defined in the protocol can streamline the process of creating the TV domain (and the eCRF for that matter). Having an unambiguous set of Visits defined in the protocol can also help avoid inconsistencies between raw data provided by third-party vendors and the eCRF data. Work to make visit definitions clear in the protocol and you will be rewarded with consistent Visit content in the raw data from all sources. In fact, why not consider defining VISITs (as needed for TV) as part of the protocol process. For that matter why not start the protocol process by drafting the trial design domains. Let’s move to the next section and see what we think about that idea. Hint: we like it a lot.

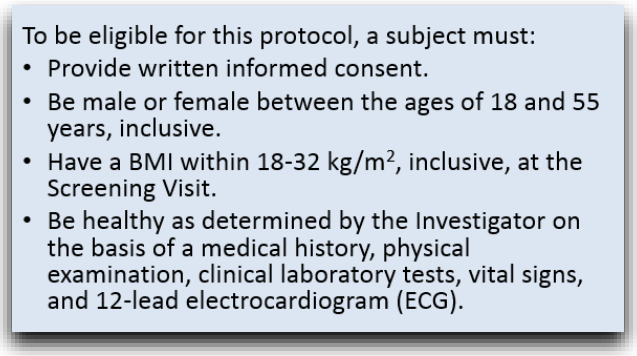
THE BEST PROTOCOLS ARE BASED ON TRIAL DESIGN METADATA

TDM domains database the protocol and summarize the study. They contain most of the key elements that form the protocol in an easily accessible format. Why not start the protocol design process with metadata designed to support TDM domains?

If you asked yourself: “How was the last TI domain I saw created?” your answer would likely be that it was cut and pasted from the protocol (or protocols if multiple criteria versions were relevant). And possibly you would add that you have a few bad memories of protocol admission criteria that needed a lot of updates before they would fit in TI. Would things have been better (for you anyway) if the TI metadata was created first and then used to populate the protocol?

Sure there are some wrinkles to parse through (most notably that often a shortened version of criteria are needed in TI and non-printable characters may enhance readability but are not welcome in submission data) but it is hard to argue against the practical advantages of having a single, programmatically accessible source for inclusion/exclusion criteria. And having procedural wrinkles such as needing criteria to be under 200 characters for TI may motivate the protocol writers to choose short options where possible and opens the door to having them create TI friendly short versions as part of protocol development (as opposed to having programmers rewrite the content as they see fit).

The advantages of using TI metadata as a first step grow when you are working through a study that has had many criteria versions due to protocol amendments. Keeping a running list (as TI metadata) by Amendment ensures the information is organized and easy to find and highlights adjustments made over the course of study conduct. Take a look the example snip of protocol inclusion criteria in figure 2 below. Can you think of any reasons why it is better to copy and paste them from a protocol into TI metadata than it is to generate the protocol criteria based on TI metadata?


To be eligible for this protocol, a subject must:

- Provide written informed consent.
- Be male or female between the ages of 18 and 55 years, inclusive.
- Have a BMI within 18-32 kg/m^2, inclusive, at the Screening Visit.
- Be healthy as determined by the Investigator on the basis of a medical history, physical examination, clinical laboratory tests, vital signs, and 12-lead electrocardiogram (ECG).

Figure 2. Sample Protocol Inclusion Criteria

Similarly, generating TS metadata as a first step in protocol development will ensure that a granular overview is firmly in hand before the writing starts and can be used to populate many sections of the protocol; generating TV metadata can inform the schedule of events and naturally focuses attention on having clear definitions for clinical encounters; generating Trial Arms (TA) and Trial Elements (TE) metadata can make it easier to ensure that the granular components (elements) of the planned study design are clearly documented (and supported by collection) and that the sequences of those elements are clearly established; and last but not least, generating Trial Disease Assessment (TD) metadata will help highlight the assessments related to efficacy where applicable.

TDM domains contain a wide range of protocol content in an easily accessible format. Why not start the protocol design process with metadata used to create TDM domains? You have to create TDM domains for submission; why not create them early so your colleagues at work can benefit from them too. And to take it one step further, why not manage trial design metadata across a submission effort. Having consistently formed trial design data across a submission will help avoid many of the hassles typically encountered when pooling data and will help the submission effort as a whole.

SDTM FRIENDLY CRF DESIGN

Let's move on to the next step in clinical data collection, generation of the collection forms (the eCRF). We are still thinking from an advertising standpoint but standards implementation is a process not and event and you have to keep your enthusiasm! What do we want? SDTM! When do we want it? Now! How can we do that? One good way is to make it so CRF design processes play nice with SDTM mapping. And yes, it is important to have a CDISC expert involved in this step. The big idea here is that we want to collect what we submit. Mapping will always be needed as collecting data in SDTM format is not a practical option; but if the collected substance is consistent with SDTM requirements, mapping challenges (and associated costs and risks) are greatly reduced.

GOOD CRFS LEVERAGE CDISC CT

Have you ever seen an eCRF pick list that just won't cooperate with CDISC CT? Hate that, right? The problem may be with the eCRF. While some picklists are specified in the protocol, many are not. And following CDISC CT when designing CRF picklists can make for an easy, seamless transition to SDTM. Figure 3 is an example of a really awkward (from an SDTM perspective) collection page. Figure 4 is the same content collected with SDTM in mind.

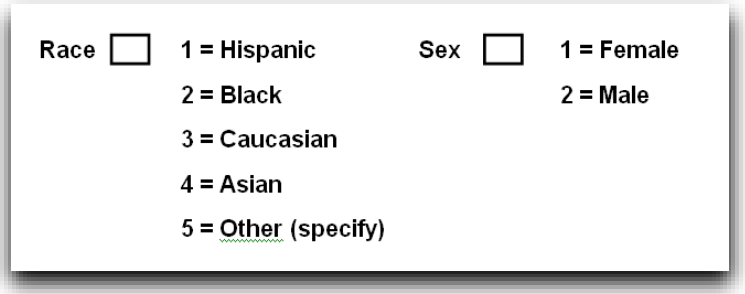


Figure 3. Not an SDTM Friendly Collection Page

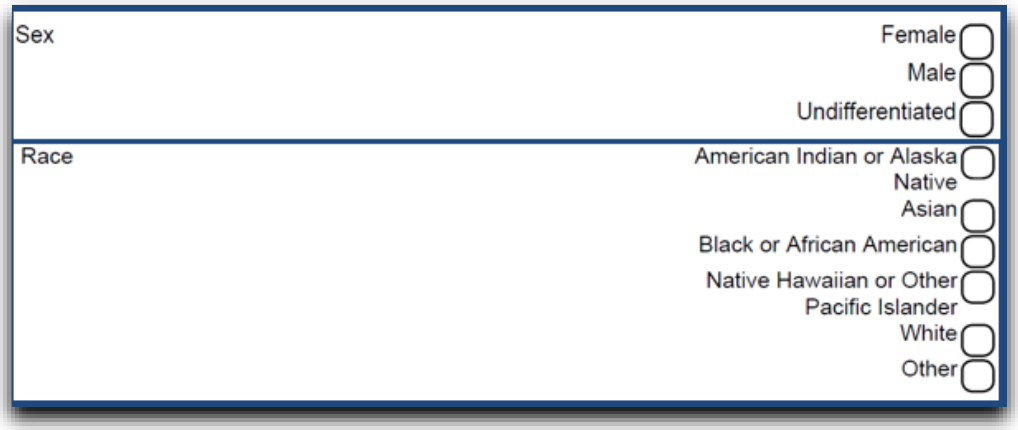


Figure 4. SDTM Friendly Collection Page

BETTER CRFS LEVERAGE CDASH

What is the easiest way to generate eCRFs that play nice with SDTM? Make use of the great work being done by the CDASH team. CDASH is not officially required. That said, it is hard to ignore the advantages of being supported by freely-available, peer-reviewed content that is based in a global standard. With CDASH forms (as an added bonus, at no cost to you) you also get raw variable names and labels.

While CDASH does not support absolutely everything, CDASH will meet most collection needs; and leveraging an approach that is consistent with CDASH is typically an easy lift when an area is not

specifically supported. For example, it is easy and useful to use CDASH naming conventions when coming up with new variable names for unsupported forms. And don't forget that new eCRFs are often available in Therapeutic Area User Guides (TAUGs). Sure you can design your own SDTM friendly eCRFs from scratch; but in the long run you have to ask yourself why you would want to.

THE BEST CRFS ALWAYS COLLECT (AND OUTPUT) THE SAME INFORMATION THE SAME WAY

Would you generate a completely new CRF from scratch for each study? Will you get a medal if you find the most unique ways to collect race on a CRF? Is it good to randomly assign raw variable names and labels? No, no and no. And it can even be a health risk: We have a confirmed report of a programmer who has nightmares about a study where the clinical data had three different ways of naming dates (--DTC, --DAT, --DT). And the –DTC versions were not fully consistent with ISO 8601. To this day he lives in fear that it will happen again.

Ensuring that the same content is collected (and delivered) in the same way (within a study eCRF and across studies) is not a simple task, but having stable inputs for SDTM mapping makes it so code can be easily used and reused. Not to mention, the study coordinators at the investigator sites will be grateful that your eCRFs don't change with every study.

PREPARING TO CONSISTENTLY IMPLEMENT SDTM WITH METADATA STANDARDS

Let's move on to the next step in the clinical data planning process, preparing to map the data as collected to SDTM. We are still thinking from an advertising standpoint (don't you all deserve the best?) but standards implementation is a process not and event so you have to keep your enthusiasm! What do we want? SDTM! When do we want it? Now! How can we do that? By supporting SDTM mapping with metadata standards.

You may have noticed that we started with planning data collection at the protocol and eCRF level. That planning (in life as in this paper) needs to be in place if you plan to generate and maintain comprehensive and useful SDTM metadata standards. If the content and format of the collected data are not stable, planning opportunities are naturally limited; if the substance of the collected data is not consistent with SDTM, SDTM implementation will be convoluted, and the final product compromised.

GOOD METADATA STANDARDS SUPPORT SDTM VARIABLES AND LABELS

Let's start with the most basic building block of SDTM mapping: Metadata that supports creation of SDTM variable names, labels and other attributes. What are those other attributes? Datatype (character or numeric), Length, Core (Required, Expected or Permissible), Role (Identifier, Topic, Synonym Qualifier, Grouping Qualifier, Result Qualifier, Variable Qualifier and Record Qualifier). Many of you will already have this level of metadata in place. Hooray! For those who do not, the good news is that this content is available from our friends at CDISC. Pretty great either way. Right? Right!

When building (or updating) SDTM variable level metadata it is useful to consider how the metadata will integrate with programming. For example, if the metadata will also serve as the starting template for study specific SDTM specifications, it makes sense to include more than the variable level content available from CDISC. Table 2 below shows some basic variable-level metadata content for VS. Of note in Table 2: A keep column is used to support limiting variables to just what is needed without removing rows from the template; all variables supported by the SDTM model for VS are included to make it so nothing will need to be added, e.g. VSREFID; variable lengths are set to the maximum allowed by CDISC rules, which works well when variables will be set to minimum length through programming in preparation for submission. Other "custom" columns such as sponsor specific mapping guidelines or default mapping based on corporate level standards should be considered.

KEEP	VARIABLE	LABEL	TYPE	LENGTH	CORE
Y	STUDYID	Study Identifier	Char	200	Req
Y	DOMAIN	Domain Abbreviation	Char	2	Req

Y	USUBJID	Unique Subject Identifier	Char	200	Req
Y	VSSEQ	Sequence Number	Num	8	Req
N	VSGRPID	Group ID	Char	200	Perm
N	VSREFID	Reference ID	Char	200	Perm
Y	VSSPID	Sponsor-Defined Identifier	Char	200	Perm
Y	VSTESTCD	Vital Signs Test Short Name	Char	8	Req
Y	VSTEST	Vital Signs Test Name	Char	40	Req
N	VSCAT	Category for Vital Signs	Char	200	Perm
N	VSSCAT	Subcategory for Vital Signs	Char	200	Perm
N	VSPOS	Vital Signs Position of Subject	Char	200	Perm
Y	VSORRES	Result or Finding in Original Units	Char	200	Exp
Y	VSORRESU	Original Units	Char	200	Exp

VARIABLE	CORE	ROLE
STUDYID	Req	Identifier
DOMAIN	Req	Identifier
USUBJID	Req	Identifier
VSSEQ	Req	Identifier
VSGRPID	Perm	Identifier
VSREFID	Perm	Identifier
VSSPID	Perm	Identifier
VSTESTCD	Req	Topic
VSTEST	Req	Synonym Qualifier
VSCAT	Perm	Grouping Qualifier
VSSCAT	Perm	Grouping Qualifier
VSPOS	Perm	Record Qualifier
VSORRES	Exp	Result Qualifier
VSORRESU	Exp	Variable Qualifier

Table 2. Example SDTM Variable Level Metadata

BETTER METADATA STANDARDS SUPPORT VALUES AND INCORPORATE CONTROLLED TERMONOLOGY

Once you have variable level metadata in place (Yea!) it is time to build on your success. Supporting value-level content and building in controlled terminology. For findings domains, --TEST and –TESTCD values make it clear what type of results are in a row. Establishing metadata that supports all possible –TEST and –TESTCD values (including any sponsor-specific additions) will help ensure values are consistently applied and (when built into a spec template) can streamline study mapping and review. The documented values can be leveraged to support study mapping (avoiding huge numbers of conditional statements in mapping notes), and they can be used to verify that mapping is consistent with CDISC CT including sponsor additions once mapping is complete. Table 3 below shows a simple example of VS value level metadata. As with the variable level content, there is a KEEP column so the values needed for a specific study can be chosen without deleting rows from the template. VSORRES and VSORRESU columns are there to support study level mapping; if corporate level collection is very stable, it is often useful to establish standard mapping in those fields.

KEEP	VSTESTCD	VSTEST	VSORRES	VSORRESU
Y	BMI	Body Mass Index	<insert mapping>	<insert mapping>
Y	BSA	Body Surface Area	<insert mapping>	<insert mapping>

Y	DIABP	Diastolic Blood Pressure	<insert mapping>	<insert mapping>
Y	HEIGHT	Height	<insert mapping>	<insert mapping>
N	HR	Heart Rate	<insert mapping>	<insert mapping>
N	OXYSAT	Oxygen Saturation	<insert mapping>	<insert mapping>
Y	PULSE	Pulse Rate	<insert mapping>	<insert mapping>
Y	RESP	Respiratory Rate	<insert mapping>	<insert mapping>
Y	SYSBP	Systolic Blood Pressure	<insert mapping>	<insert mapping>
N	TEMP	Temperature	<insert mapping>	<insert mapping>
N	WEIGHT	Weight	<insert mapping>	<insert mapping>

Table 3. Example SDTM Value Level Metadata

This same general approach should be applied wherever it is useful to drill down and assign content by variable value. An additional case is with QNAM and QLABEL. As supplemental variables are (for the most part) sponsor defined, it is even more critical to document the values as there is no general reference. Also worth noting is that at times two or three (or more) variables are needed to document when certain values are assigned. For example, LBCAT, LBSPEC and LBMETHOD may be needed in addition to LBTEST and LBTESTCD to define a unique test. The general concept of value level metadata is not limited to findings domains. It is often useful to document mapping within a variable or variables whenever a variable value determines the type of content to be included in other variables (such as within DSTERM).

Drilling down a bit more, supporting mapping to CDISC and sponsor CT at the variable value level can (and should) be supported by metadata standards. When mapping is based in collection standards that are used across studies it is often best to build the associations between the collected content and what is needed in SDTM into globally available metadata standards. Congratulations if you are thinking that this level of metadata should not be needed as (with proper planning) the collected information would match what is needed for SDTM. Great point! But while we all work to ensure data collection seamlessly integrates with SDTM, it is important to prepare for any data points that do not. And there are always cases (such as clinical labs, especially local labs) where the raw test names are entered as free text fields. In those cases, mapping metadata will need to be updated on a regular basis to stay current.

On to some examples. Table 4 below is an example of mapping metadata for lab test units. It is shocking how many ways there are to designate 10^9/L (even if you ignore typos). Table 5 below shows an example of LBTEST and unit mapping metadata that includes (as an added bonus, free to all readers!) conversion factors. For isolated fields that do not match what is needed in SDTM, metadata to support mapping to SDTM or sponsor CT is often best supported at the study level. Table 6 below shares an example of AEACN mapping metadata.

RAW_UNIT	LBORRESU
10^3/CMM	10^9/L
10^3/MCL	10^9/L
10^3/MM^3	10^9/L
10^3/MM3	10^9/L
10^3/UL	10^9/L

Table 4. Example SDTM Unit Mapping Metadata

LBTESTCD	RAW_TEST	LBTEST	LBORRESU	CONV_FACT	LBSTRESU
PHOS	Inorganic Phosphate	Phosphate	mEq/L	3.1	mg/dL
PHOS	Phosphorous	Phosphate	mg/dL	1	mg/dL
PHOS	PHOSPO	Phosphate	mg/L	0.1	mg/dL
PHOS	Phosphorus	Phosphate	mmol/L	3.1	mg/dL

Table 5. Example LBTEST and LBSTRESU Mapping Metadata

AEACN_CRF	AEACN
Dosing Interrupted	DOSING INTERRUPTED
None	DOSE NOT CHANGED
Permanently Discontinued	DRUG WITHDRAWN
Reduced	DOSE REDUCED

Table 6. Example AEACN Mapping Metadata

THE BEST METADATA STANDARDS FULLY SUPPORT DATA AND DEFINE.XML

I bet you are thinking “Wow, metadata can do almost everything!” But wait, there’s more! Have you ever scrambled to generate a define.xml close to a delivery or submission deadline? One common reason for that is that is that study level mapping metadata typically are not designed to fully support generation of the define.xml. If you have not already done it, a few simple adjustments to can make it so specifications gracefully support define.xml generation.

It is likely that you already have most of the information you need for the define.xml in your specifications template already. The two key items that are often not supported by local standards - but needed for generation of a define.xml - are the source of a variable or value (Assigned, CRF, Derived, eDT or Protocol) and a plain language comment field that is used to house descriptions for derived and assigned variables. Spec writers know where variables and values come from and can use their first-hand knowledge of how content is derived and assigned to generate associated define.xml comments. Including a place for that information (and supporting it with general guidelines on how to populate) takes advantage of the information spec writers have in hand and ensures that the define.xml is supported early so it can be efficiently generated when needed.

Table 7 below shares a snip of a specification that contains variable level content including origins and comments. Where derivation comments repeat themselves across domains and/or studies it is often worthwhile to spend the time to establish standard text. As CRF page references are not possible until the SDTM annotated CRF (acrf.pdf) is completed, a separate column is provided for page numbers to facilitate adding page references after the origins are populated and updating references if a new version of the eCRF is released.

DOMAIN	VARIABLE	LABEL	ORIGIN	PAGE	DEFINE_COMMENT
LB	EPOCH	Epoch	Derived		Derived based on TA/TE using LBDTC. Date comparisons are done using the highest common level of precision.
LB	LBDTC	Date/Time of Specimen Collection	CRF	22	

LB	LBDY	Study Day of Specimen Collection	Derived		Study day of collection in relation to first exposure to study medication (RFSTDTC). RFSTDTC is considered to be Day 1.

Table 6. Example Variable Level Specifications with Origin and Comment

While it is reasonable to prepare for times when the acrf.pdf is not available when designing a specification template, there are many advantages to generating it early. In addition to supporting define.xml origin references, the acrf.pdf provides a useful way to understand data mapping (everyone likes pictures, few willingly read specifications) and even (with a bit of coding or a vendor application) can be used to directly populate CRF origin information. And to go a step further, the purpose of data documentation is to make it easier for data consumers (such as the FDA) to use the data. Why not make the documentation available when the data are first mapped? Your local data consumers will appreciate it!

DESIGNING ROBUST, REUSABLE MAPPING CODE

Let's move on to the next step in the clinical data planning process, generating the code used to map the data as collected to SDTM. We are still thinking from an advertising standpoint (don't you all deserve the best?). How can you make sure you have the best? By supporting SDTM programming with standard processes and useful tools. Please keep in mind that whenever we mention specifications below, we are assuming that the specs completely and accurately describe what programming needs to accomplish.

Any competent programmer can read specifications and generate an SDTM domain. Any two (competent, independent) programmers working from the same spec will generate exactly the same domain. However, unless each is supported by standard programming processes and tools, the path they take to the result will not be as efficient and robust as it could be. Supporting programmers with standard code saves time, eliminates risk and allows your talent to focus on the study-specific details that can't be prospectively supported.

It should go without saying by now (but we will anyway): it is not possible to have efficient SDTM mapping code unless you have (wait for it) stable, SDTM friendly data inputs. And how can you make that happen? You already know this. By doing the upstream work needed to ensure that protocols and eCRFs play nice with SDTM and are consistently constructed.

GOOD MAPPING CODE LEVERAGES METADATA

Mapping specifications (and recall we are assuming complete and accurate mapping specifications based on sponsor metadata standards are used) have a wealth of information that should be leveraged. Specifications contain variable names, labels and other attributes as needed to support efficient programming. Better to read in and programmatically use that metadata than to type it into a program as separate step.

At a basic level, domain and variable names, labels (and other attributes) should be generated based on metadata. The spec snip in Table 2 above would work nicely as an input for this. Just in case you want an example of how to turn variable specifications into SAS® variable attributes, the code below will work nicely:

```
%macro create(domain);
data specs;
        set pullin.SDTM_STRUCTURE; **** SAS data version of specs;
        attrib newlength length = $8;
        newlength = trim(left(length));
        if keep = 'Y';
        if upcase(datatype)='TEXT' then do;
        newlength = '$'||trim(left(newlength));

data sdtm;
```

```
proc sql noprint;

select variable into :retain separated by ' '
        from specs
where dataset="&data.";

select n(variable) into :nvar
        from specs
where dataset="&data.";

select variable into   :var1-:var%left(&nvar)
        from specs
where dataset="&data.";

select label into :des1-:des%left(&nvar)
        from specs
where dataset="&data.";

select newlength into :len1-:len%left(&nvar)
        from specs
where dataset="&data.";

quit;

data SDTM_FINAL;
        set SDTM;
        %do i=1 %to &nvar.;
                attrib &&var&i.. label="&&des&i.." length=&&len&i..;
        %end;
%mend create;
```

Going a step further, mapping raw content for consistency with SDTM CT is best done with metadata. With this approach, the code does not need to be adjusted if the specification changes. Remember Tables 4, 5 and 6 above. The example mapping metadata that targeted controlled terminology? It is fine if you want to go back and look. We will wait for you.

And don't forget that the TDM domain content can be used to support other domains. Nervous that IETEST will not exactly match between TI and IE? Why not populate IE.IETEST based on TI.IETEST? Wondering how to simply ensure that scheduled visits will be consistent across multiple domains? How about using the TV specification to support VISIT mapping across domains. It won't solve every issue but is a nice step forward. Sample TV mapping metadata is provided below in Table 7. Is it clear how this could be used to support VISIT mapping across domains? It sure is!

VISITNUM	VISIT	VISITDY	TVSTRL	TVENRL	eCRF Folder	eCRF FolderSeq
1	Screening		Start of Screening element.	The day before the start of the first treatment element	SCR	-14
2	Day 1	1	The day of the first treatment element		D1	1
3	Day 8	8	8 days (+/-1) after the start of the first treatment element		D8	8

Table 7. Example TV Specifications

BETTER MAPPING CODE IS SUPPORTED BY STANDARD TOOLS AND PROCESSES

Different programmers program (well) differently. And while a many different programming paths will lead to the correct answer, a clear path from collection to SDTM should be established when designing SDTM programming processes.

Just as you have a standard header template for programs, it can make sense to set up a standard flow for SDTM programs. For each domain, the same steps need to be completed. Referencing those steps in a starter template for code helps ensure that all steps are completed and can serve as a useful prompt for standard practice items. Sample high level process steps are below:

- Pull in all raw data sources
- Prep raw data as needed for downstream use
- Generate variable attributes based on metadata (remember this?)
- Run general tools (see below)
- Use metadata to map to controlled terminology (remember this?)
- Sort by key variables, assign xxSEQ
- Keep and order needed variables, label domain (based on metadata), and output as an xpt file

When you will need to do the same (or similar) mapping again and again it makes sense to support that mapping with standard code. How much detail can usefully be included is largely a function of how stable the raw data inputs are. Very stable inputs allow for comprehensive mapping support; if inputs are variable, default mapping support is necessarily limited to generally applicable items.

General tools (most often macros) handle routine SDTM mapping tasks. Structural data variability across domains (and studies) is accounted for with macro variables and judicious use of data prep. One simple example is a macro or macros to handle converting raw dates (from varying collection formats) to ISO 8601 date variables in SDTM. A more complicated example is a macro used to assign EPOCH based on a comparison of the relevant date from the domain and reference dates generally documented in the TDM TA and TE domains and metadata.

THE BEST MAPPING CODE IS EFFICIENT, ROBUST AND REUSABLE

You deserve the best, right? Right! How great would it be to have a standard program for each domain that could be used with minor or no modifications? Really great right? Right! Assuming (and this is a big assumption) that the inputs are SDTM friendly and stable across studies you can have just that. As a disclaimer, there will always be the need to adjust mapping metadata and to tweak some macro references for study-specific content such as VISITs. But that said, when data are stable and SDTM-friendly, code can be easily reused.

Having a stable set of raw data inputs (supported by good metadata standards and useful macros for repeating situations) is the biggest part of the battle, but not all of it. The best laid data plans are still subject to noise in the form of dirty and incomplete data; and even if everything in the collected data are perfectly as planned, there may be content collected in free text fields that is subject to CT. To efficiently monitor these types of issues, it helps to have code that flags observations that violate the assumptions on which the code is based. Often these are data issues that will resolve. Flagged observations can also indicate that mapping metadata is incomplete or part of the code is not sufficiently versatile, offering something akin to a continuous improvement process at both the study and the global level.

Once mapping code (including flagging of potential issues) is in place for a study, it is likely that it will need to be run on a monthly (or weekly or daily) basis to meet the needs of all data consumers. Having a process in place to automatically generate the SDTM domains as needed and report any unexpected complications adds efficiency to your process.

CONCLUSION

You deserve the best, right? Right! Over the course of this paper we have suggested some practical ways that you (yes you, we still expect each of you to do your part) can help build a faster track for SDTM. Anyone reading this a visual learner? Just like pictures? What follows in Figure 4 is a pictorial overview of the process steps that we have discussed.

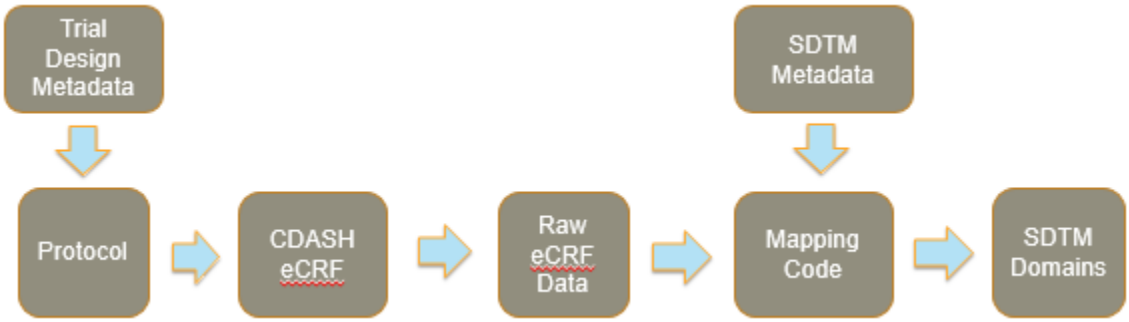


Figure 4: Mapping Process Overview

With good planning in place, SDTM domains can be efficiently generated across studies and the data made available as needed to all the downstream users who depend on it.

ACKNOWLEGMENTS

Thanks FDA, CDISC working groups, and our present and past companies for providing the opportunity to learn this stuff, as well as their continued support.

REFERENCES

CDISC web site. Study Data Tabulation Model (SDTMIG). V3.2. http://www.cdisc.org/

Study Data Technical Conformance Guide. U.S. Department of Health and Human Services, FDA, CDER, CBER. https://www.fda.gov/downloads/ForIndustry/DataStandards/StudyDataStandards/UCM384744.pdf

Dunn, Shelley, Codelists Here, Versions There, Controlled Terminology Everywhere. PharmaSUG 2016. http://www.pharmasug.org/proceedings/2016/DS/PharmaSUG-2016-DS16.pdf

Sullivan, Sue. SDTM Metadata: The Output is only as Good as the Input. PharmaSUG 2016. http://www.lexjansen.com/pharmasug/2016/PO/PharmaSUG-2016-PO15.pdf

Abolafia, Jeff and DiIorio, Frank. Protocol Representation: The Forgotten CDISC Model. PhUSE 2016. http://www.lexjansen.com/phuse/2016/cd/CD01.pdf

van Bakel, Bas. DIY: Create your own SDTM mapping framework. PhUSE 2016. http://www.lexjansen.com/phuse/2016/cd/CD03.pdf

Goud, Judith and Shetty, Priya. How a Metadata Repository enables dynamism and automation in SDTM-like dataset generation. PhUSE 2016. http://www.lexjansen.com/phuse/2016/dh/DH05.pdf

Kelly, Kristin, et al. SDTM TE, TA, and SE Domains: Demystifying the Development of SE. PharmaSUG 2015. http://www.pharmasug.org/proceedings/2015/DS/PharmaSUG-2015-DS02.pdf

Salyers, Jerry, et al. Considerations in Creating SDTM Trial Design Datasets. PharmaSUG2014. http://www.pharmasug.org/proceedings/2014/DS/PharmaSUG-2014-DS03.pdf

Ewing, Daphne. Automating SDTM File Creation: Metadata Files Speeding the Process. SAS Global Forum 2010. http://support.sas.com/resources/papers/proceedings10/181-2010.pdf

Bradford J. Danner, Exploitation of Metadata for Restructuring Datasets and Code Generation. PharmaSUG 2008 , http://www.lexjansen.com/pharmasug/2008/cc/CC21.pdf

Abolafia, Jeff and DiIorio, Frank. Managing The Change and Growth of a Metadata-Based System. SAS Global Forum 2008. http://www2.sas.com/proceedings/forum2008/128-2008.pdf

CONTACT INFORMATION

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

Steve Kirby
Chiltern
Steven.Kirby@Chiltern.com

Mario Widel
Eli Lilly & Co.
mwidel@Lilly.com

Richard Addy
Chiltern
Richard.Addy@Chiltern.com