**PhUSE 2013**

**Paper DH05**

# Going Against the Flow: Backmapping SDTM Data

Pantaleo Nacci, Novartis Vaccines & Diagnostics Srl, Siena, Italy

## ABSTRACT

Novartis Vaccines has embarked on a multi-year project to remap all available legacy study data from a proprietary standard to an industry-standard one. But while these legacy data are inspected and mapped to SDTM in a validated way, there are still urgent activities requiring the pooling of data mostly in the old standard. This paper talks about the experience gathered while developing a generic program to back-map data from the new NVD standard to the previous one.

## INTRODUCTION

Many pharma companies are already in the process of remapping all available legacy data to a common data standard, usually one or more of the CDISC ones (SDTM/ADaM), or are planning to do it sooner rather than later.

Novartis Vaccines is one of them, and this activity has now been running for more than two years as part of the much wider Clinical Data Repository (CDR) project. The initial timelines were both aggressive and optimistic, so that the planned period during which data would have been split between the two standards was extremely short. In fact this period proved to be much longer, due to a combination of scarce experienced resources, competing activities and legacy data issues.

To both keep my SAS skills up to speed and keep options open, I decided to develop a program which could back-map data originally collected in CDASH and transformed into SDTM 3.1.2 to the old, pre-CDISC standard. This program is currently unvalidated, so it is not used in production runs.

## FROM LEGACY SYSTEMS TO CDR

As I wrote elsewhere, in NVD we have available legacy data in electronic format from as far as 1992 (and we have indications that further archaeological work might uncover stuff even older than that). Over the years data have been managed, stored and analysed using a variety of platforms and solutions: from single MS-DOS PCs, to Vax VMS systems, to Windows Server 2003 ones, and now SAS Drug Development (SDD).

Throughout all these phases, the data standard used in Chiron before and in Novartis Vaccines later has remained remarkably stable, if we exclude the initial couple of years when data management and analysis were performed by external CROs, each using their own methods. This stability helped a lot the remapping exercise, since it was often possible to create highly reusable routines.

When we decided to move from the classical setup (i.e., a remote server with SAS installed on it as an application) to a modern integrated environment (Figure 1), the opportunity to create at the same time a real repository for all data ever collected became all of a sudden both very appealing and technically feasible. The number of identified legacy studies was around 400, covering a span of at least 20 years, and until now data from less than 200 of them have been already remapped.

As soon as they have been remapped to the flavour of SDTM 3.1.2 used in NVD and stored in their own study-specific directory, all data are automatically added to a so-called ‘ocean’; during the pooling process further harmonisation tasks are performed, e.g., all adverse events are recoded to the same version of MedDRA.

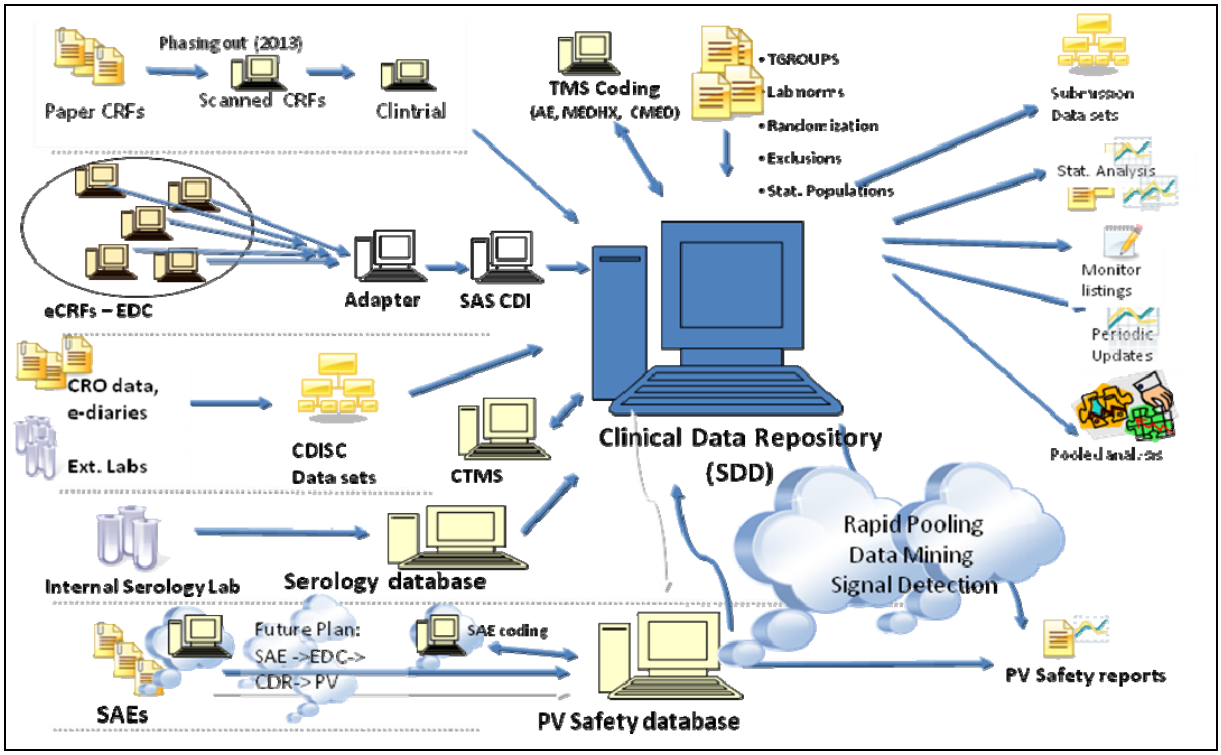


Figure 1: The CDR as an integrated environment for multiple processes

## ANALYSING DATA DIRECTLY FROM THE OCEAN

The initial plan stipulated that once all data for a certain vaccine/adjuvant/product were in the ocean, this should be used as the only data source for any analysis spanning more than one study, like DSURs/PSURs, IBs, etc. But, there is always a 'but'…

The actual legacy data remapping is taking much longer than expected, for several reasons. The result is a hybrid situation, where for important products most of but not all the data are in CDR, while several others are lingering behind at various stages of porting.

While we wait for all data to be remapped, several approaches have been devised by the programmers according to the task at hand, possibly including a one-off on-the-fly remapping to SDTM of just the data strictly needed for a task, e.g., for a DSUR only a subset of demographic and safety data are needed.

As a proof of concept, and as a further option, I decided to develop a SAS program doing exactly the opposite, i.e., taking as input the data in the new CDASH/SDTM format and mapping them back to the legacy standard: this might be helpful when the majority of the data for a vaccine haven't been ported to SDTM yet.

As a side note, one of the most interesting features of SDD is the ability to manage several data standards in an integrated way, even if in NVD we chose not to take advantage of it: by default SAS Institute provides and supports the various CDISC ones, but there is no reason why a company could not set up the environment with the definition of their own.

### LOOKING FOR (A FULL DESCRIPTION OF) THE LEGACY STANDARD

As in all such exercises, the first thing needed was a full description of the target standard, which in this special case was the legacy rather than the new one.

As I wrote above, the Chiron/NVD data standard remained remarkably stable over time since its inception in the mid-'90s, but this doesn't mean that there were no changes:

- Some variables went from numeric to character, or the other way round (e.g., month part of dates)
- A few variable were used in different studies to store different information
- Different variables were used to collect the same info
- Several versions of non-standard panels were designed and used over the years, e.g., those to collect influenza-like illness (ILI) data
- For some reason the implementation of EDC led to existing variables being sometimes defined with unexpected lengths: while not a real issue in itself, this fact leads to a lot of warnings in the SAS log, stating that strange things might be happening to the data when pooling them
- Then early EDC CDASH studies, where data were remapped to the old standard within Clintrial, led to more of the same warnings, since most variables ended up with a length of 200 by default

I ended up selecting what I considered the best representation of the standard structure for each panel from different studies, so I got, e.g., DEMOG and CBP from one study, ADVERSE and CMED from another, and so on:

```
* Get empty dataset structures ;
libname in ('!FLU\V71_10S\FINAL\PROD\SSD'
            '!FLU\V70_25S\FINAL\PROD\SSD'
            '!FLU\V71_27S\FINAL\PROD\SSD') access=readonly;

%macro skelet(ds=);
data &ds._;
  set in.&ds (where = (prot = "&snumber")
%if %upcase(&ds)=POSTINJ %then %do;
              rename = (swel_uni = swelunit)
%end;
              );
run;
%mend;
%skelet(ds=adverse);
%skelet(ds=cbp);
%skelet(ds=cmed);
%skelet(ds=complian);
<etc.>
```

The macro `&snumber` contains the current study code, so the `where` statement creates empty data sets, while the `%if` statement is used to fix one of the several small issues with the legacy standard, in which naming conventions were not always implemented consistently (Figure 2).

| | | | | | | |
|---|---|---|---|---|---|---|
| 25 | IRED_CRF | Character | 10 | $10. | $10. | INJ SITE STUDY VACCINE ERYTHEMA ON CRF |
| 26 | IREDUNIT | Character | 2 | $2. | $2. | INJ SITE STUDY VACCINE ERYTHEMA UNIT |
| 27 | IREDD | Numeric | 8 | | | INJ SITE STUDY VACCINE ERYTHEMA (MM) |
| 28 | IHARD_CR | Character | 10 | $10. | $10. | INJ SITE STUDY VACCINE INDURATION ON CRF |
| 29 | IHARDD | Numeric | 8 | | | INJ SITE STUDY VACCINE INDURATION (MM) |
| 30 | IHARDUNI | Character | 2 | $2. | $2. | INJ SITE STUDY VACCINE INDURATION UNIT |
| 31 | CHILLS | Character | 2 | $2. | $2. | CHILLS |
| 32 | CHILLSW | Character | 4 | | | CHILLS |
| 33 | MALAISE | Character | 2 | $2. | $2. | MALAISE |
| 34 | MALAISEW | Character | 4 | | | MALAISE |
| 35 | MYALGIA | Character | 2 | $2. | $2. | MYALGIA |
| 36 | MYALGIAW | Character | 4 | | | MYALGIA |
| 37 | FATIGUE | Character | 2 | $2. | $2. | FATIGUE |
| 38 | FATIGUEW | Character | 4 | | | FATIGUE |
| 39 | ARTHRALG | Character | 2 | $2. | $2. | ARTHRALGIA |
| 40 | ARTHRALW | Character | 4 | | | ARTHRALGIA |
| 41 | QUICK | Character | 1 | $1. | $1. | QUICK BOX |
| 42 | QUICKW | Character | 16 | | | QUICK BOX |
| 43 | HEADACHE | Character | 2 | $2. | $2. | HEADACHE |
| 44 | HEADACHW | Character | 4 | | | HEADACHE |
| 45 | ECCH_CRF | Character | 10 | $10. | $10. | MEASURED ECCHYMOSIS ON CRF |
| 46 | ECCHUNIT | Character | 2 | $2. | $2. | ECCHYMOSIS UNIT |
| 47 | ECCHD | Numeric | 8 | | | ECCHYMOSIS IN MM |
| 48 | SXFEVER | Character | 2 | $2. | $2. | FEVER POST INJECTION |
| 49 | SXFEVERW | Character | 7 | | | FEVER POST INJECTION |
| 50 | SWEL_CRF | Character | 10 | $10. | $10. | SWELLING ON CRF |
| 51 | SWEL_UNI | Character | 2 | $2. | $2. | SWELLING UNIT |
| 52 | SWELD | Numeric | 8 | | | SWELLING DERIVED IN MM |

Figure 2: Inconsistencies in variable naming conventions

**HOW WELL DEFINED IS THE NEWER STANDARD ANYWAY?**

The other side of the same coin, even if issues tend to be less pronounced in that case, was the variability in our own CDASH/SDTM implementations: I use the plural because at least for SDTM we ended up having two of them. The first one was designed early, then used to plan and implement Phase 1 of the Legacy Data Conversion (LDC), while

the other, different in several subtle ways, was the one which was actually used for the new studies, set up, run and analysed completely in CDR. So for example we realised that compliance phone call data should have been included in DS only after Phase 1 of the remapping was completed.

#### CODE EXAMPLE: GENERAL MACROS

During the development of the program I wrote a couple of general-use macros, one to combine supplemental domains (SUPP--) with their main ones, whenever available. The logic used to check the existence of a non-empty data set was as follows:

```
%macro combine_supp(domain=);
%if %sysfunc(exist(sdtm.&domain)) %then %do;
  %let dsid = %sysfunc(open(sdtm.supp&domain));
  %if &dsid %then %do;
    %let isobs = %sysfunc(attrn(&dsid,any));
    %let rc = %sysfunc(close(&dsid));
  %end;
  %else %let isobs = 0;
  %if %sysfunc(exist(sdtm.supp&domain)) & &isobs=1 %then %do;
    <etc.>
  %end;
%end;
%mend;
%combine_supp(domain=ae);
...
```

The other one dealt with rebuilding date and time variables starting from ISO 8601 values (--DTC):

```
%macro get_date(in=, out=);
  if &in.dtc ^= '' then do;
    &out.dy = input(scan(&in.dtc, 3, '-T'), 2.);
    &out.mo = scan(&in.dtc, 2, '-');
    &out.yr = input(scan(&in.dtc, 1, '-'), 4.);
    &out.dt = mdy(max(input(&out.mo, 2.), 1), max(&out.dy, 1), &out.yr);
    if &out.mo ^= '10' then &out.mo = compress(&out.mo, '0');
    if index(&in.dtc, 'T') > 0 then &out.tm = input(scan(&in.dtc, 2, 'T'), time5.);
  end;
%mend;
```

#### CODE EXAMPLE: IDENTIGYING THE RIGHT PANELS FOR RELREC RECORDS

The RELREC domain posed a particularly difficult problem, since its records had to be merged into multiple panels according to several factors, according to a quite complex algorithm.

This is an excerpt where all the interested SDTM domains are identified:

```
proc sort data = relrec
          out = relrec_;
  by usubjid relid rdomain;
run;

data relrec_ (keep = studyid usubjid relid condensed);
  set relrec_;
  condensed = compress(rdomain || '|' || idvar || '|' || idvarval);
run;

proc transpose data = relrec_
               out = relrec_2 (drop = relid _name_);
  by studyid usubjid relid;
  var condensed;
run;

data aece (drop = vars vars_ col3-col7)
     aecm (drop = vars vars_)
     aecmmh (drop = vars vars_)
     cmmh (drop = vars vars_);
  set relrec_2;
  length col1-col7 $ 602;
```

```
  vars = scan(col1, 1, '|') || scan(col2, 1, '|') || scan(col3, 1, '|') ||
scan(col4, 1, '|') || scan(col5, 1, '|') || scan(col6, 1, '|') || scan(col7, 1,
'|');
  vars_ = tranwrd(vars,  'AEAE', 'AE'); vars_ = tranwrd(vars,  'MHMH', 'MH');
  vars_ = tranwrd(vars_, 'AEAE', 'AE'); vars_ = tranwrd(vars_, 'MHMH', 'MH');
  vars_ = tranwrd(vars_, 'AEAE', 'AE'); vars_ = tranwrd(vars_, 'MHMH', 'MH');
  vars_ = tranwrd(vars_, 'AEAE', 'AE'); vars_ = tranwrd(vars_, 'MHMH', 'MH');
  vars_ = tranwrd(vars_, 'AEAE', 'AE'); vars_ = tranwrd(vars_, 'MHMH', 'MH');
  select(vars_);
    when('AECE') output aece;
    when('AECM') output aecm;
    when('AECMMH') output aecmmh;
    when('CMMH') output cmmh;
  end;
run;
```

The repeated TRANWRD statements are a simple if not particularly elegant way to remove useless duplicates.

**CODE EXAMPLE: DATA FORMATS**

Variable codes used in the two standards were of course quite different, but central repositories for the standard codelists exist in both cases, so rather than using IF or SELECT statements, cumbersome to maintain and difficult to reuse, I decided to recode variables as much as possible using a two-step approach: first recreating the coded value using temporary formats, then using the central format library to populate the human-readable version (identified by a -W suffix):

```
proc format;
  ...
  value $vyn
    'N' = '2'
    'Y' = '1'
    'NA' = '3'
    ;
run;

data adverse;
  set ae;
  ...
  serious = put(aeser, $vyn.);
  seriousw = put(serious, $yesnos.);
  ...
run;
```

**CODE EXAMPLE: DATA COLLECTED IN CDASH, BUT NOT PRESENT IN SDTM**

So-called administrative variables (e.g., 'Did the subject report any AEs?') are in general now allowed in SDTM, so I had to go back to the CDASH data sets to get them.

```
* Get full PRESTUDY and CONT info from CDASH data ;
data cm_c (drop = studyoid subjid _temp_);
  set cdash.cm (keep = studyoid subjid cmspid cmongo cmprior
                rename = (cmspid = _temp_)
                where = (_temp_ ^= .));
  length studyid usubjid cmspid $ 200;
  studyid = studyoid;
  usubjid = compress(scan(studyid, 1, 'EX') || '-' || subjid);
  cmspid = strip(put(_temp_, best4.));
run;
```

**CODE EXAMPLE: DEATH DATA**

There is no domain for death data in SDTM 3.1.2, so it was unclear where the relevant data might be found. I ended up using both AE.AESDTH (a variable which BTW was invariably left empty in legacy studies) and DS.DSTERM.

```
data death_add1 (keep = prot ptno deathdy deathmo deathyr deathdt deathtm causesp);
  length prot $ 18
         ptno $ 7;
  set ae (where = (aesdth = 'Y'));
  prot = studyid;
```

```
  ptno = scan(usubjid, 2, '-');
  %get_date(in=aest, out=death);
  causesp = aeterm;
run;

data death_add2 (keep = prot ptno);
  length prot $ 18
         ptno $ 7;
  set ds (where = (upcase(dsterm) = 'DEATH'));
  prot = studyid;
  ptno = scan(usubjid, 2, '-');
run;
```

Additional details for the one death reported in CDR studies until today and present in DS only, are stored as free text in CO, and as such not usable programmatically.

#### CODE EXAMPLE: CDISC NON-STANDARD VARIABLES

Since the current goal of the program is to recreate selected panels in the old standard specifically for pooling purposes, I chose not to remap most non-standard variable. Anyway I structured the program in such a way that the addition of new panels would be quite simple, once the target structure is fully defined.

E.g., in a couple of studies I needed to recreate the panel of physical examination data. In that panel a simple binary variable is present, collecting if at a certain visit a planned physical examination had been performed or not: I had to look in two different CDASH domains to find the same information, since in one study it was collected in LB (LBPHYASS), in another in EX (EXPHYASS).

```
%if %sysfunc(exist(cdash.ex)) %then %do;
  %let dsid = %sysfunc(open(cdash.ex,i));
  %let varexist = %sysfunc(varnum(&dsid,EXPHYASS));
  %let rc = %sysfunc(close(&dsid));
  %if &varexist>0 %then %do;
    <etc.>
  %end;
%end;
...
%if %sysfunc(exist(cdash.lb)) %then %do;
  %let dsid = %sysfunc(open(cdash.lb,i));
  %let varexist = %sysfunc(varnum(&dsid,LBPHYASS));
  %let rc = %sysfunc(close(&dsid));
  %if &varexist>0 %then %do;
    <etc.>
  %end;
%end;
```

#### CODE EXAMPLE: SOLICITED ADVERSE EVENT DATA

The reshaping of the local and systemic reactions data, also known as solicited adverse events, typical of allergology and vaccine studies, was a good exercise in the use of macro language. On paper CRFs these data were collected and stored by time point, so each record in the database contained the values for all observations performed at a certain time (‘fat’ structure). On the contrary, in eCRFs we adopted a ‘skinny’ approach, more flexible, so that each record refers to just one result at a certain point in time. As an additional complicating factor there are reactions collected in mm, others as presence or severity, according to the age of the subjects, then body temperature and finally a few which are not really reactions.

These are just a couple of snippets from a much longer subroutine dealing with these data:

```
%let sev_list = IPAIN ARTHRALG CHILLS FATIGUE HEADACHE DIARRHEA INAPPET MFEVER
                MALAISE MYALGIA NAUSEA PRURITIS SWEAT VOMIT RASH
                PTENDSV EATCHSV IRRITSV SLEEPSV;
%let pre_list = ANALGESI ANAANTP ANAANTT;
%let mea_list = ECCH IHARD IRED SWEL;

data postinj_;
  length
%do i = 1 %to %sysfunc(countw(&sev_list));
         %scan(&sev_list, &i) $ 2
%upcase(%sysfunc(compress(%sysfunc(subpad(%scan(&sev_list, &i), 1, 7))))w) $ 8
%end;
```

```
%do i = 1 %to %sysfunc(countw(&pre_list));
         %scan(&pre_list, &i) $ 2
%upcase(%sysfunc(compress(%sysfunc(subpad(%scan(&pre_list, &i), 1, 7))))w) $ 8
%end;
%do i = 1 %to %sysfunc(countw(&mea_list));
         %upcase(%substr(%scan(&mea_list, &i)unit, 1, 8)) $ 2
         %upcase(%substr(%scan(&mea_list, &i)_crf, 1, 8)) $ 10
%end;
    ;
  set postinj_;
run;
...
data vs_postinj;
  set vs (where = (scan(vstpt, 1) = 'DAY' & vsstat ^= 'NOT DONE'));
  select(vstestcd);
    when('TEMP') do;
      temp = input(vsorres, best8.); tempunit = upcase(vsorresu);
      tempuniw = tempunit; tempd = round(vsstresn, 0.1); temploc = vsloc;
      visnum = visitnum;
    end;
  end;
run;
...
```

I used SUBPAD rather than SUBSTR to avoid messages in the log about variables being too short.

## CONCLUSION

The possible benefits of having all study data in one place and in a common, well defined format can be many and covering a wide range of applications, but to be able to reap them the remapping period should be kept as short as possible: starting the data porting at full steam only to stop or even slow down mid-way is a dangerous proposition, since there is a real risk of creating more confusion rather than a clearer situation. Needing to guess on which of two platforms an analysis will run with the least problems is not a good thing, and this is the situation in which we find ourselves for some of our products. On the plus side the legacy data conversion tasks are finally getting momentum again, and the most critical gaps should be closed by the end of the year, so that all data for selected products will finally be available in CDR.

My backmapping program was born as a proof of concept and is still a living project, since almost every time a new study goes live in CDR a new slight variation of something requires a change in the program. This change might be just the physiological addition a new LBTESTCD value, easily addressable, or the appearance of a brand new domain, requiring full setup of the corresponding panel of the legacy standard. Until now I have maintained only one version of the program dealing with all studies, so keeping a validated status for it once attained would be very difficult: as a matter of fact this happened for one study, data from which were urgently needed to complete a DSUR. The validation approach adopted in that occasion was quite original, since the fastest way we could think of was to have another programmer map again the remapped data to SDTM, checking that at the end of the complete circle data were the same.

## ACKNOWLEDGMENTS

I would like to thank Andy Ndikom, who suggested that this program might be a good starting point to write a paper.

## CONTACT INFORMATION

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

Pantaleo Nacci
Novartis Vaccines & Diagnostics Srl
Via Fiorentina, 1
53100 Siena, Italy
Phone: +39 0577 243554
Fax: +39 0577 278443
Email: pantaleo.nacci@novartis.com
Web: www.novartisvaccines.com