PharmaSUG 2014 - Paper AD17

The New Tradition: SAS® Clinical Data Integration

Vincent J. Amoruccio, Alexion Pharmaceuticals, Inc., Cheshire, CT
Janet Stuelpner, SAS, Cary, NC

ABSTRACT

Base SAS® programming has been around for a very long time. And over that time, there have been many changes. New and enhanced procedures, new features, new functions and even operating systems have been added. Over time, there have been many windows and wizards that help to more easily generate code that can be used in programs. Through it all, programmers always come back to their SAS roots, the basic programs with which they started. But, as we move into the future, is this the best use of time, to sit and manually code everything? Or can we take advantage of the new tools and solutions that generate code and use metadata to describe data, validate output and document exactly what the programmer has done. This paper will show you how we can change the current process using the graphical user interface of SAS Clinical Data Integration to integrate data from disparate data sources and transform that data into industry standards in a methodical, repeatable, more automated fashion.

INTRODUCTION

When SAS® Institute, Inc. incorporated in 1976, it was a statistical analysis system used primarily for data management, statistics and reporting. As demand for this type of software grew, and the hardware and software industry evolved, SAS software changed. It evolved from a single to a multi-platform software and ultimately became accessible on all platforms. It adopted a friendlier user approach that mirrored the graphical user interface of other computing environments such as Macintosh and Windows. It expanded into multiple industries across the globe and is now used at more than 70,000 sites in 139 countries. Needless to say, SAS is now the leader in analytic software and services.

Despite the evolution of SAS for more than 35 years, the primary purpose of SAS for Clinical and Statistical programmers across the world has been to retrieve, transform, analyze, and report on data. The reason we use SAS has remained the same but how we program has changed. As SAS programmers, we always come back to the old tradition of writing Base SAS® code by hand. In *'Confessions of a Clinical Programmer: Dragging and Dropping Means Never Having to Say You're Sorry When Creating SDTM Domains'*, we learned that manually writing Base SAS code might not be the most effective use of our time. We were introduced to a new tool known as SAS® Clinical Data Integration ("CDI") which could generate code dynamically and use metadata to describe, validate, and document SAS programs. Because of CDI's graphical user interface ("GUI"), we learned that programming could be easier, faster, and much more efficient than Base SAS programming.

SAS is the standard for clinical trials data and electronic submissions to regulatory authorities across the world, such as the FDA. The healthcare and pharmaceutical industry, one of the most regulated industries, have also seen a fair share of changes in regulation over the last 35 years. In '*The Y2K17 Bug! Using Metadata to Respond to PDUFA V Requirements*', we learned about U.S. regulations such as the Food and Drug Administration Act ("FDASIA") and the Prescription Drug User Fee Act ("PDUFA"). We learned about a new law signed by President Obama that would require the FDA to expedite applications and sponsors to use the standards from the Clinical Data Interchange Standards Consortium ("CDISC") by the year 2017. We went to our old traditions and use Base SAS and metadata to create CDISC Study Data Tabulation Model ("SDTM") and Analysis of Dataset Model ("ADaM") output.

Since the publishing of '*Confessions of a Clinical Programmer: Dragging and Dropping Means Never Having to Say You're Sorry When Creating SDTM Domains'* three things have changed. First, CDI has evolved further and can now be used to create ADaM output. Second, sponsors are now required to adopt, use, and submit data using CDISC standards. Finally, the submission process in the U.S. has been expedited. As a result, sponsors will need to create CDISC compliant output faster.

The purpose of this paper will be to show how CDI can produce CDISC compliant output more efficiently and faster than Base SAS programming. What did the programmer do in the past to create CDISC compliant output? What can be gained from CDI? What is done now in CDI to make the process more efficient and faster than Base SAS programming? What is the future of our new tradition? From the old techniques of Base SAS programming to the new tools of CDI, we will show how CDI better supports the pharmaceutical industry needs for transforming, managing, and verifying the creation of CDISC Standards. As a result, we will explain why CDI is the new tradition for clinical and statistical programmers.

THE SOURCE

The examples that we will use for this paper are based on the same data from *'Confessions of a Clinical Programmer: Dragging and Dropping Means Never Having to Say You're Sorry When Creating SDTM Domains'*. The study is a blinded control study in which 902 subjects were randomly assigned to receive either a single intravenous dose of Nicardipine Hydrocholoride (Nicardipine) or placebo. Nicardipine is a calcium channel blocker, considered for the treatment of patients who have had a particular type of stroke classified as aneurismal subarachnoid hemorrhage (SAH). This type of stroke occurs when an aneurysm bursts inside the brain.

We used the Demographics (DM) and Exposure (EX) SDTM domain to create a sample ADaM Subject-Level Analysis Dataset (ADSL). The ADSL dataset contains one record per subject. Its primary purpose is to provide required variables from the ADaM Implementation Guide (IG) as well as other subject-level variables to describe a subject's experience in a trial. The ADSL is minimally required when submitting clinical trial data to the Food and Drug Administration (FDA) even if no other analysis datasets are submitted.

The SDTM data we used is old legacy data which used a version older then SDTM version 3.1.2. As a result, some variable names and types are different. You can find a sample of these data in the Appendix. The Nicardipine data did not have ADaM data, so the ADSL was created specifically for this paper using ADaM 2.1. It contains all of the required variables described in the ADaM IG and a few additional variables.

THE OLD TRADITION

Industry Standards & Requirements

Since its formation in 1997, the Clinical Data Interchange Standards Consortium (CDISC) has been an open, multi-disciplinary, neutral non-profit organization working to develop standards to streamline medical research. CDISC gained notoriety in July, 2004 when acting FDA Commissioner Dr. Lester Crawford announced the FDA's intent to receive data in a standard format known as the Study Data Tabulation Model (SDTM). Enforcing a standard format such as SDTM would increase the efficiency of reviewing and evaluating data and get approved drugs into the public's hands faster.

Despite Dr. Crawford's profound intent, the message was interpreted as a suggestion rather than a requirement. FDA reviewers are still spending inordinate amounts of time reorganizing and restructuring data received in varying formats. After roughly a decade, new drug applications (NDA) and biological license applications (BLA) are not getting approved quickly and drugs are not getting into the public's hands any faster today than they were in 2004. This is because the FDA's 2004 statement lacked enforceability. It was a suggestion, not a requirement!

In July 2012, eight years after Dr. Crawford's message, President Obama delivered a message much more penetrating to the pharmaceutical industry than Dr. Crawford's. He signed into law the Food and Drug Administration Innovation Act (FDASIA). As part of FDASIA, he also reinstated the Prescription Drug User Fee Act for the fifth time (PDUFA V). FDASIA and PDUFA V made three significant changes that directly impact organizations involved in medical research such as pharmaceuticals. First, it mandated ICH M2 EWG Electronic Common Technical Document Specifications. By doing so it eliminated paper submissions to the FDA and enforced electronic submissions. Second, it established a new review model that will improve the efficiency and effectiveness of the FDAs review process. Ultimately, it will decrease the number of review cycles required for approval of NDAs and BLAs. Finally, it explicitly states that CDISC terminology and standards will be required.

With each authorization of PDUFA, the FDA communicates goals and procedures of the reauthorization for the fiscal years following its release. As expected the FDA released a document titled "PDUFA Re-authorization Performance Goals and Procedures Fiscal Years 2013 Through 2017" to provide a more intimate explanation of the PDUFA V requirements. The FDA followed up to this document by releasing a position statement in September 2013 about study data standards for submissions. This statement reinforced the requirement of CDISC standards for submissions and announced final guidance to be expected shortly.

CDISC is changing the landscape of submissions in the United States, but what about other countries? Is CDISC global? While there is no explicit language from regulatory agencies outside of the U.S. like Japan's Pharmaceutical and Medical Device Agency (PMDA) and the European Medicine Agency (EMA), there is strong evidence to suggest to that a CDISC requirement is imminent. The PMDA began collaborating with and following CDISC in 2013. The EMA released an expectation paper for electronic source data and data transcribed to electronic data collection tools in clinical trials in 2010. No matter how you spin it, CDISC is a requirement in the U.S., and an acceptable, imminent requirement globally.

The BASE SAS Approach

In '*The Y2K17 Bug! Using Metadata to Respond to PDUFA V Requirements*' we learned how to create and use a system of metadata to define attributes about the domains, the variables, and the parameter values. In this section,

we will use a similar technique to show how to implement the ADaM data standard to create the ADSL dataset using BASE SAS as your primary tool. This process defines metadata about the ADSL using a Microsoft® Excel spreadsheet to define the domain and variable attributes. Since ADSL contains one record per subject, there are no parameters to define. The spreadsheet will be converted into SAS datasets using a metadata import program. It will be applied separately through a SAS macro in the program that sets up and creates the ADSL.

The metadata for the ADSL is shown in the appendix. Converting the metadata into SAS datasets involves importing the Excel spreadsheet into BASE SAS, transforming that data with DATA steps and SQL, and then saving the metadata as permanent SAS datasets.

The SAS code for our example of importing the metadata into SAS from the Excel spreadsheet looks like this:

```
libname adammeta "&_datapath\ADAM\METADATA";
libname inxls pcfiles
       server = "&compname..alxn.net"
       port = 8621
       path = "&_datapath\ADaM\Metadata\adam-database-specifications.xlsx";

proc sql;
       create table all_sheets1 as
       select distinct upcase(scan(memname,1,'$')) as sheet
       from Sashelp.Vtable
       where upcase(libname) = "INXLS" and
             index(memname, "$") gt 0 and
             (substr(upcase(memname),1,3) = "DOM" or substr(upcase(memname),1,2) =
              "AD");
quit;

data all_sheets2;
       set all_sheets1;

       if sheet = "DOMAINS" then int=1;
              else if sheet = "ADSL" then int = 2;
              else int = 3;
run;

proc sort data = all_sheets2;
       by int sheet;
run;

data all_sheets_final;
       set all_sheets2;
       order = _n_;
run;

data _null_;
       set all_sheets_final end = last;
       i+1;

       call symput('sheet'||strip(put(i,best.)), cats(sheet));
       if last then call symput('total', strip(put(i,best.)));
run;

data domains;
       length DomainLabel $40. RepeatingYN $3.;
       set inxls."Domains$"n;

       DomainLabel = Domain_Label;
       DomainKeys = Domain_Keys;
       RepeatingYN = Repeating___Y_N__;

       if Domain ne '';
run;
```

```
proc sql;
       create table adammeta.domains as
       select Domain, DomainLabel, DomainKeys, Purpose, RepeatingYN, Structure
       from domains
       order by Domain;
quit;

%macro sheets;
%do i = 2 %to &total;

data &&sheet&i (rename=(domain = _domain_));
       length VariableName $8. VariableLabel $40. VariableType $4. VariableCategory
       $40. SASFormat $25. SubmitYN $3.
                     ;
       set inxls."&&sheet&i$"n;

       VariablePosition = Variable__Position;
       VariableName = Variable_Name;
       VariableLabel = Variable_Label;
       VariableType = Variable__Type;
       VariableCategory = Variable__Category;
       VariableLength = Variable__Length;
       SASFormat = SAS_Format;
       SortOrder = input(Sort__Order,best.);
       SubmitYN = Submit___Y_N__;

       %if &&sheet&i = ADSL %then %do;
       InAllDataSetsYN = In_All__Data_Sets___Y_N__;
       %end;

       if VariableName ne '';
run;

proc sql;
       create table adammeta.&&sheet&i as
       select _domain_ as domain length = 8, VariablePosition, VariableName,
       VariableLabel, VariableType, VariableCategory, VariableLength, SASFormat,
       SortOrder, SubmitYN, core,
       %if &&sheet&i = ADSL %then %do;
       InAllDataSetsYN,
       %end;
       Definition
       from &&sheet&i
       order by VariablePosition;
quit;
%end;
%mend sheets;
%sheets;
libname inxls clear;
```

Creating a macro to apply the metadata involves reading in the correct metadata, joining the domain level and variable level metadata together, and then transforming the data using DATA step and SQL. This process dynamically create a temporary dataset and macro variables that will be used later to apply the dataset and variable metadata.

The SAS code to create a macro to apply the metadata looks like this:

```
%macro attradam;

%global listall listalls datalbl orderby listc listn listnn total numcnt charcnt dom
        username compnm varflag mem memcnt;

%let listc=dummy;
%let listn=dummy;
```

```
%let listnn=dummy;

%let username = %sysget(username);
%let compnm   = %sysget(computername);

proc sql feedback noprint;
       select strip(value) into : dom
       from Sashelp.Vmacro
       where name = "PGMNAME";
quit;
%let dom = %upcase(&dom);
%let dm = %str(&pgmname.);

proc sql feedback;
       create table &dom.1 as
       select a.Domain,
              a.VariablePosition, a.VariableName, a.VariableLabel, a.VariableType,
              a.VariableLength, a.SASFormat, a.SortOrder, b.DomainLabel
       from adammeta.&pgmname as a
            left join
            adammeta.domains as b
       on a.domain = b.domain
       where upcase(a.SubmitYN) = "YES";
quit;

%if &dom ne ADSL %then %do;
proc sql;
       create table adsl_vars as
       select *
       from adammeta.adsl
       where upcase(InAllDataSetsYN) = "YES";
quit;

data &dom.1a;
       set adsl_vars (in = sl)
       &dom.1 (in = dom)
            ;
       if sl then neworder = 1000;
       if dom then neworder = 2000;
run;

%end;

%if &dom eq ADSL %then %do;

data &dom.1a;
       set &dom.1;
       neworder = 1;
run;

%end;

data &dom.2(keep=length VariableName VariableLabel VariableType VariableLength
                 SortOrder SASFormat VariablePosition charcnt numcnt)
       sortord1  (keep=VariableName SortOrder neworder)
       position1 (keep=VariableName VariablePosition neworder);

       set &dom.1a (rename=(SASFormat=SASFormatz));
       length SASFormat $14;
       retain charcnt numcnt 0;

       if SASFormatz>'' and index(SASFormatz,'.')=0 then
       SASFormat=cats(SASFormatz)||'.';
```

```
      if SASFormatz>'' and index(SASFormatz,'.')>0 then SASFormat=cats(SASFormatz);

      if SASFormat = '' then SASFormat = "X";

      if propcase(VariableType)='Char' then
            length=compress('$'||put(VariableLength,4.));
            else length = strip(put(VariableLength,4.));

      if DomainLabel > '' then call symput('datalbl',strip(DomainLabel));

      if propcase(VariableType) = 'Char' then charcnt+1;
      if propcase(VariableType) = 'Num' then numcnt+1;
      call symput('charcnt',put(charcnt,3.));
      call symput('numcnt',put(numcnt,3.));
run;

proc sort data=sortord1;
      by neworder SortOrder;
run;

data sortord;
      set sortord1;

      if NewOrder > . and SortOrder > . then NewSortOrder = NewOrder+SortOrder;
            else NewSortOrder = . ;
      if NewSortOrder>.;
run;

proc sort data=position1;
     by neworder VariablePosition;
run;

data position;
      set position1;

      if NewOrder > . and VariablePosition > . then NewPosition =
            NewOrder+VariablePosition;
            else NewPosition = . ;
      if NewPosition >.;

      if upcase(VariableName) = "STUDYID" then NewPosition = 1;
      if upcase(VariableName) = "USUBJID" then NewPosition = 2;
      if upcase(VariableName) = "SUBJID" then NewPosition = 3;
      if upcase(VariableName) = "SITEID" then NewPosition = 4;
run;

proc sql feedback noprint;
      select VariableName into :listc separated by ' '
      from &dom.2
      where propcase(VariableType) = 'Char';

      select VariableName into :listn separated by ' '
      from &dom.2
      where propcase(VariableType) = 'Num' and numcnt<30;

      select VariableName into :listnn separated by ' '
      from &dom.2
      where propcase(VariableType) = 'Num' and numcnt>29;

      select VariableName into :listall separated by ', '
      from position
      order by NewPosition;
      %let listall = &listall;
```

```
       select VariableName into :orderby
       separated by ', '
       from sortord
       order by NewSortOrder ;
quit;

%let listalls= %sysfunc( tranwrd(%nrbquote(&listall),%str( ,),%str( ) ) );

data _null_;
       set &dom.2 end=last;
       i+1;
       call symput('var'||strip(put(i,best.)), strip(VariableName));
       call symput('lbl'||strip(put(i,best.)), cats(VariableLabel));
       call symput('lgt'||strip(put(i,best.)), strip(length));
       call symput('fmt'||strip(put(i,best.)), strip(SASFormat));

       if last then call symput('total', strip(put(i,best.)));
run;

options quotelenmax;

data final;
       %do i = 1 %to &total.;
       attrib
       &&var&i. length = &&lgt&i. label = "&&lbl&i"
              %if &&fmt&i. ne X %then %do;
                     format = &&fmt&i.
              %end;
       ;
       %end;

       %if &listc. gt and &charcnt>0 %then %do;
              array listc{*} $ &listc.;
              do j = 1 to dim(listc);
                     listc{j}=' ';
              end;
       %end;

       %if &listn. gt and &numcnt>0 %then %do;
              array num{*} &listn.;
              do k = 1 to dim(num);
                     num{k}=.;
              end;
       %end;

       %if &listnn. gt and &numcnt>28 %then %do;
              array numm{*} &listnn.;
              do k = 1 to dim(numm);
                     numm{k}=.;
              end;
       %end;

       if _n_ = 1 then delete;
run;

proc sql feedback;
       create table &pgmname._attrib as
       select &listall.
       from final;
quit;

options quotelenmax;
```

```
proc datasets lib=work;
     delete final path &dom.2 sortord sortord1 position position1;
quit;

%mend attradam;
```

In this section, we explore how to put bring in the metadata and create ADSL. For this particular data, it simply involves joining the SDTM DM and EX data together to obtain all of the variables necessary to derive ADSL variables.

Since there was only one constant dosing interval per subject, the EX domain is one record per subject. There are also no supplemental qualifiers available. The transformation of data is as simple as two DATA steps, two SQL and one APPEND procedure. A sample of the ADSL can be found in the Appendix.

The SAS code to create the ADSL looks like this:

```
%attradam;

proc format;
       value $ sex
       'M' = '1'
       'F' = '2';
run;

data dm;
       set sdtm.dm;

       rfstdt = input(rfstdtc, e8601dt.);
       sexn = input(put(sex, $sex.),best.);

       if race = "ASIAN" then racen = 1;
              else if race = "BLACK OR AFRICAN AMERICAN" then racen = 2;
              else if race = "WHITE" then racen = 3;
              else if race = "OTHER" then racen = 4;

       brthdt = input(brthdtc, yymmdd10.);
       endtc = rfendtc;

       trtsdt  = input(scan(rfxstdtc, 1, 'T'),e8601da.);
       trtstm  = input(scan(rfxstdtc, -1, 'T'),e8601tm.);
       trtsdtm = input(rfxstdtc, e8601dt.);

       trtedt  = input(scan(rfxendtc, 1, 'T'),e8601da.);
       trtetm  = input(scan(rfxendtc, -1, 'T'),e8601tm.);
       trtedtm = input(rfxendtc, e8601dt.);

run;

proc sql;
       create table combined1 as
       select a.*, b.extrt
       from dm as a
              left join
            sdtm.ex as b
       on a.usubjid = b.usubjid;
quit;

data final;
       set combined1;

       if extrt = "IV NICARDIPINE" then do;
              fasfl = 'Y'; * Full Analysis Set Population Flag *;
              saffl = 'Y'; * Safety Population Flag *;
              iitfl = 'Y'; * Intent-To-Treat Population Flag *;
       end;
```

```
      else if extrt = "PLACEBO" then do;
            fasfl = 'N'; * Full Analysis Set Population Flag *;
            saffl = 'N'; * Safety Population Flag *;
            iitfl = 'Y'; * Intent-To-Treat Population Flag *;
      end;

      array setc (3)$ fasfl saffl iitfl;
      array setn (3)  fasfn saffn iitfn;

      do i = 1 to 3;
            if setc(i) = 'Y' then setn(i) = 1;
            else if setc(i) = 'N' then  setn(i) = 0;
      end;
run;

%let listvar=%sysfunc( tranwrd(%nrbquote(&listall),%nrbquote(,),%str( )) );

proc append base=&dom._attrib
           data=final(keep=&listvar) force;
run;

proc sql feedback;
      create table adam.&dom (label = "&datalbl.") as
      select &listall
      from  &dom._attrib
      order by &orderby;
quit;
```

Advantages

There are many advantages to using BASE SAS. First and foremost, it is relatively inexpensive. The only resources required are you, your computer, and BASE SAS software. If you are going to use remote files like Microsoft Excel or Microsoft Access for your target metadata, you will also need the SAS Access Interface to PC files.

Second, there are no restrictions when using BASE SAS and there is no setup time. As previously mentioned in *"Confessions of a Clinical Programmer: Dragging and Dropping Means Never Having to Say You're Sorry When Creating SDTM Domains",* you have a full armory of procedures and functions at your disposal to transform the data as you please. It allows you to be creative and innovative with your code from the beginning and will not force you into using any one particular procedure or function.

Finally, BASE SAS provides you the ability to develop and test your code iteratively instead of running the code in large jobs. This provides for easy error detection and repair.

Disadvantages

There are many disadvantages with the BASE SAS approach. While the software license fee for CDI is larger than the fee for BASE SAS, There is a large amount of time involved in the setup of the BASE SAS approach when metadata is used. In this approach, the metadata is setup and managed by the user, not the software. This can lead to increased costs for the human resources required to manage the process and the code.

Using metadata, as we have defined it in this section, will definitely improve your ability to create SDTM or ADaM datasets, however, it is still prone to human error, inconsistencies, and cumbersome custom code. In our example, there are four sources of data and code required to produce a single dataset. This creates the potential issues with portability of code from one trial to another and management of code.

Finally, and most importantly, you cannot create a fully submittable and CDISC compliant submission using BASE SAS alone. There are no pre-existing utilities that allow you check for compliance with CDISC, such as OpenCDISC or WebSDTM. There are no pre-existing utilities to create define.xml or define.pdf files with hyperlinks to annotated case report forms and transport files. You can integrate the SAS Clinical Standards Toolkit with BASE SAS to perform these missing tasks but, there is more time, code, and manipulation involved to do so.

Business Justification for CDI

Base SAS programming has been around for almost four decades. CDISC standards have been around for almost two decades. During this time many pharmaceutical companies have used various methods to transform clinical data from its raw source into a standard format. Using BASE SAS has been a traditional and verifiable method but has

many disadvantages associated with it. While there have been many enhancements to these traditional methods, each project is still unique and requires expensive manual coding that is neither standard nor repeatable.

What if there was a solution that streamlined the transformation of clinical data, reduced overhead costs associated with maintaining manual, custom code and the time to complete a fully compliant regulatory submission? Would companies be willing to spend more money upfront on such a tool if it allowed for downstream efficiencies? Would companies be willing to retire traditional processes using BASE SAS and implement a new solution that generates code and uses metadata to describe data, validate output and document exactly what the programmer has done in a more efficient and timely manner?

Yes! Many pharmaceutical companies are turning to SAS Clinical Data Integration as the new tradition for the transformation of clinical data into standard, compliant, and submittable data. Even the FDA has turned to SAS Clinical Data Integration as a new way of reviewing data. At the Institute of Medicine's 2012 workshop on sharing clinical data research, Ron Fitzmartin, Senior Advisor in the Office of Planning and Informatics at FDA's Center for Drug Evaluation and Research (CDER), used SAS CDI to demonstrate mapping clinical trial data into the SDTM format. He said "using this tool, reviewers do not have to spend time piecing data together and can quickly drill down to the patient-level data to look at outliers and elevated values".

There are many similar tools available from other companies. So, why choose SAS over other competitors? First and foremost SAS has been the leader in business analytics software and services since 1976. SAS has almost 14,000 employees worldwide. SAS is used by the FDA and more than 45,000 companies in 100 countries. SAS has a verifiable ability to offer depth of resources to sustain product development and support. Its competitors are smaller, in fewer countries and have not been around nearly as long. SAS' competitors have designed similar tools with different interfaces however, the underlying software is still SAS software. It's no wonder the majority of SAS' customers continue to use them.

THE NEW TRADITION

What if you could use a common platform to aggregate all the sources of clinical trials data across the entire drug life cycle – from end to end – so that critical decisions are based on complete, current information and handoffs occurred seamlessly with no corruption or loss of data? A single integrated platform could do that for you. Using Base SAS is a very manual process. What if you could automate and make repeatable data processing tasks that could free your resources to concentrate on the tasks that require special attention and expertise? A solution that has repeatable processes could do that for you. What if compliance with CDISC standards is built into the system, so that as data is processed, you could submit the results of your trial to the regulatory authorizes quickly? If the CDISC standards were embedded standards into a solution you could. What if you could analyze the structure and content of your data so that you can address inconsistencies, compliance with the standards and data quality issues proactively, before potential problems impact your study? If you had the capability of using metadata impact analysis you could.

SAS Data Integration Studio (DI Studio) provides a visual way through a graphical user interface (GUI) for building, implementing and managing data integration processes no matter what the source of the data, application, or platform. In 2009, a solution based on DI Studio was created to incorporate clinical metadata so that users can transform the raw clinical trials data into industry standard format. Using the SAS Clinical Standards Toolkit (CST), the extract, transform and load (ETL) process for clinical data was born. It is an easy to use and manage, multiple-user environment that allows the use of repeatable processes that are easily shared. The creation and management of data and metadata are improved with extensive impact analysis of potential changes made across all data integration processes.

SAS Clinical Data Integration (CDI) enables its users to easily and quickly build and edit data integration processes, to automatically capture and manage standardized metadata (both clinical and non-clinical) from any source, and to display, visualize, and understand metadata and data integration processes. In addition, the clinical component allows you to check your standardized output for compliance to CDISC standard domains, create the define.xml and create the transport files for your regulatory submissions all with simple transformations that were built for this purpose.

Automatic Code Generation: SAS Clinical Data Integration

One of the things that has made programming in Base SAS code the method of choice is the great variability in how data is transformed from its raw format to the industry standards. Give ten programmers the same task to do and there could be at least 10 different ways in which it is programmed. The variability of the way that people code is also its greatest vulnerability causing the need for duplicate programming to validate the results. This causes a great deal of extra work because you want to make sure that the validation programmers are not able to preview the code that was written by the clinical programmer. Even if the programmers are siloed and not capable of reviewing each other's work, the same assumptions and interpretations can result in an incorrect data in the domains.

CDI is a code generator. Through pre-coded transformations, tasks are put together to create a SAS program. The code is visible and can be reviewed for each transformation as well as the total program. Whether it is joining datasets together with a PROC SQL, changing the format of the data from horizontal to vertical with a PROC TRANSPOSE or summarizing data with a PROC FREQ, there is a transformation to perform the task. It is not 'black box' programming where everything is hidden. The GUI interface helps to see the flow of the program and gives the user a visual representation of the tasks that are being done.

Using Metadata to Drive Programs

How do you trace a piece of information from its source in a clinical trial to the analysis that is in the clinical study report or the end product? In the pharmaceutical world, we call this end to end traceability. At the beginning of the use of CDISC as a standard, many companies used spreadsheets to collect the metadata for its studies. These spreadsheets were used as requirements documents to tell the programmers what derivations were needed for the analysis and from where the information came. CDI is a format system with metadata management capabilities. This allows you to accomplish many tasks including traceability of your data should a question come from a regulatory authority. Using CDI and its metadata, allows you to create mappings, perform impact analysis for potential changes, check the compliance to the CDISC standards, create tables, listings and figures (TLF) and create define.xml documents for submission. The availability of a metadata driven solution allows greater reusability and greater efficiency. A highly re-usable system reduces the time and effort to validate the quality of the data that is output from the system. Using metadata allows the users to answer questions more quickly and react fast when a regulatory authority wants to know more information.

Advantages

The key advantage of using SAS Clinical Data integration is the ability to use and manage the metadata. There is a bit of set-up that needs to be done in the solution in terms of defining the target metadata at the beginning. SAS Clinical Data Integration gives you the process flow view and the ability to drag and drop transforms (pre-written code). SAS Clinical Data Integration manages your metadata for your SDTM work. It controls the target SDTM and ADaM metadata so compliance with a defined SDTM and ADaM standard is built into your workflow. It also connects the metadata across your SDTM and ADaM domain creation so that you can analyze your data for changes and updates and also propagate a change across your SDTM data creation.

If you use SAS Clinical Data Integration as intended with standard transforms, it essentially enforces a bit of consistency of process in creating SDTM and ADaM domains. This consistency along with the process view allows for SDTM and ADaM domain creation jobs to be more easily maintained and also allows for reuse of jobs. SAS Clinical Data Integration also allows for "typical" SDTM and ADaM generation tasks, such as study day (--DY) or ISO date (--DTC) creation, to be standardized into user written transforms that can be dragged and dropped into future SDTM jobs.

SAS Clinical Data Integration provides an expansive list of transformations for you to choose from. Several of those are extremely handy in terms of creating SDTM and ADaM domains and those include the sort, transpose, data joiner, lookup table, data extraction, transport file creation and data loader transformations. Finally, SAS Clinical Data Integration is integrated with the SAS Clinical Standards Toolkit. There are pre-existing SAS CDI transformations that allow you to validate your SDTM and ADaM datasets based on the SDTM or ADaM metadata and also to automatically generate your define.xml file for both types of domains which is a huge benefit.

Disadvantages

The first challenge to using SAS CDI is the old school mentality that programmers say: 'I can do it better and faster by hand'. Like people who translates everything from one language to another before they can think in the new language, programmers need to become accustomed to a new process. It is very tempting to write a bunch of customer code, but this should be avoided as much as possible. Although previously written programs can be imported into the solution, it is wise to try to work within the system to take advantage of the metadata and how the task of validation is made easier by relying on the metadata. Another issue is that a great number of people who have been programming for many years prefer to use a DATA step. They like the flexibility and ease of use of DATA step programming. SAS Clinical Data Integration is based on PROC SQL. Some find this way of programming limited and sometime difficult to use. The initial set-up of CDI is based on a great deal of business decisions that need to be made. This will involve all members of the clinical team to decide the best way to set up a study as well as what the new roles will be in business process. One of the greatest challenges is relying on the solution to do a great deal of the work for you.

CONCLUSION

For decades, SAS programmers have been cranking out code, building applications and providing the very best that they could to move the process of analyzing clinical data toward submission. Processes have been developed over

time and new technology has helped in the endeavor to submit the results of clinical trials. As an industry, we have come to discover how standards help to make things easier and how the metadata from those standards can be used to increase traceability, increase reusability and track what happens when changes are made.

We began this paper by explaining how SAS has evolved over time, from a typical fourth generation computer language to a solution oriented one. We also explained how widely accepted industry standards, such as CDISC, have evolved from a needed entity of unenforceable practices into standards that are required by regulatory authorities around the world. Both BASE SAS and SAS CDI offer advantages and disadvantages in the process of transforming clinical data. Many companies will continue to use the tried and true old tradition of BASE SAS programming while others will adopt the new tradition of SAS CDI that offers a GUI interface, metadata management and the ability to create reusable, shareable transformations and code. This purpose of this paper was to enlighten you on how to change and enhance the current process and revolutionize the world of clinical data integration.

REFERENCES

Amoruccio, Vincent J. May 2013. “The Y2K17 Bug! Using Metadata to Respond to PDUFA V Requirements.” *The Pharmaceutical SAS User’s Group. PharmaSUG*. Chicago, Illinois.

Stuelpner, Janet, et al. May 2011. “Confessions of a Clinical Programmer: Dragging and Dropping Means Never Having to Say You’re Sorry When Creating SDTM Domains.” *The Pharmaceutical SAS User’s Group. PharmaSUG*. Nashville, Tennesse.

Institute of Medicine (US). Sharing Clinical Research Data: Workshop Summary. Washington (DC): National Academies Press (US); 2013 Mar 29. 5, Standardization to Enhance Data Sharing. Available from: http://www.ncbi.nlm.nih.gov/books/NBK137818/

ACKNOWLEDGMENTS

I would like to thank Janet for the opportunity to work with a renowned Clinical SAS Programmer. I would also like to thank Daniel Amoruccio for his unbounded support in helping make my contributions to this paper successful.

This paper would not have been written if it hadn’t been for the support given to me by my husband, Robert Stuelpner and the confidence shown to me by Vince. Bob diligently read this paper in order to correct obvious errors and keep me on the right track. He sat quietly while Vince and I worked out the bugs and reviewed the direction that we wanted this paper to take. His criticisms were constructive and his support never ending.

CONTACT INFORMATION

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

Name:	Vincent J. Amoruccio	Janet Stuelpner
Enterprise:	Alexion Pharmaceuticals, Inc.	SAS Institute, Inc.
Address:	7 McKee Place,	326 Old Norwalk Road
City, State ZIP:	Cheshire, CT 06401	New Canaan, CT 06840
E-mail:	AmoruccioV@alxn.com	Janet.Stuelpner@sas.com

APPENDIX

Sample SDTM DM Data

Obs	Study Identifier	Domain Abbreviation	Unique Subject Identifier	Subject Identifier for the Study	Study Site Identifier	Investigator Identifier	Sex	Race	Country
1	NIC001	DM	NIC001-011-001	001	011	1	M	WHITE	USA
2	NIC001	DM	NIC001-011-002	002	011	1	M	WHITE	USA
3	NIC001	DM	NIC001-011-003	003	011	1	M	WHITE	USA
4	NIC001	DM	NIC001-011-004	004	011	1	F	WHITE	USA
5	NIC001	DM	NIC001-011-005	005	011	1	F	WHITE	USA
6	NIC001	DM	NIC001-011-006	006	011	1	F	WHITE	USA
7	NIC001	DM	NIC001-011-007	007	011	1	M	WHITE	USA
8	NIC001	DM	NIC001-011-008	008	011	1	F	WHITE	USA
9	NIC001	DM	NIC001-011-009	009	011	1	F	WHITE	USA
10	NIC001	DM	NIC001-011-010	010	011	1	F	WHITE	USA

Obs	Subject Reference Start Date/Time	Subject Reference End Date/Time	Date/Time of First Study Treatment	Date/Time of Last Study Treatment	Study Day of Collection
1	2007-10-12T23:25:00	2007-10-24T10:00:00	2007-10-12T23:25:00	2007-10-24T10:00:00	1
2	2007-10-14T16:30:00	2007-10-21T10:30:00	2007-10-14T16:30:00	2007-10-21T10:30:00	3
3	2007-11-10T18:00:00	2007-11-23T19:00:00	2007-11-10T18:00:00	2007-11-23T19:00:00	2
4	2007-12-02T06:10:00	2007-12-14T18:00:00	2007-12-02T06:10:00	2007-12-14T18:00:00	3
5	2007-12-08T10:15:00	2007-12-19T00:30:00	2007-12-08T10:15:00	2007-12-19T00:30:00	2
6	2007-12-15T21:00:00	2007-12-27T20:00:00	2007-12-15T21:00:00	2007-12-27T20:00:00	2
7	2007-12-30T16:15:00	2008-01-12T16:15:00	2007-12-30T16:15:00	2008-01-12T16:15:00	2
8	2008-01-09T11:35:00	2008-01-22T08:00:00	2008-01-09T11:35:00	2008-01-22T08:00:00	2
9	2008-02-24T21:30:00	2008-03-08T00:00:00	2008-02-24T21:30:00	2008-03-08T00:00:00	2
10	2008-03-01T21:00:00	2008-03-13T19:40:00	2008-03-01T21:00:00	2008-03-13T19:40:00	1

Obs	Date/Time of Birth	Age	Age Units	Planned Arm Code	Description of Planned Arm	Actual Arm Code	Description of Actual Arm
1	1988-10-24	19	YEARS	PLA	PLACEBO	PLA	PLACEBO
2	1928-07-20	80	YEARS	NIC15	NICARDIPINE .15	NIC15	NICARDIPINE .15
3	1961-01-03	47	YEARS	NIC15	NICARDIPINE .15	NIC15	NICARDIPINE .15
4	1935-09-18	73	YEARS	PLA	PLACEBO	PLA	PLACEBO
5	1927-07-10	81	YEARS	PLA	PLACEBO	PLA	PLACEBO
6	1954-08-11	54	YEARS	PLA	PLACEBO	PLA	PLACEBO
7	1964-03-27	44	YEARS	NIC15	NICARDIPINE .15	NIC15	NICARDIPINE .15
8	1951-11-28	57	YEARS	NIC15	NICARDIPINE .15	NIC15	NICARDIPINE .15
9	1942-06-25	66	YEARS	NIC15	NICARDIPINE .15	NIC15	NICARDIPINE .15
10	1963-04-06	45	YEARS	PLA	PLACEBO	PLA	PLACEBO

Sample SDTM EX Data

Obs	Study Identifier	Domain Abbreviation	Unique Subject Identifier	Sequence Number	Name of Actual Treatment
1	NIC001	EX	NIC001-011-001	1	PLACEBO
2	NIC001	EX	NIC001-011-002	1	IV NICARDIPINE
3	NIC001	EX	NIC001-011-003	1	IV NICARDIPINE
4	NIC001	EX	NIC001-011-004	1	PLACEBO
5	NIC001	EX	NIC001-011-005	1	PLACEBO
6	NIC001	EX	NIC001-011-006	1	PLACEBO
7	NIC001	EX	NIC001-011-007	1	IV NICARDIPINE
8	NIC001	EX	NIC001-011-008	1	IV NICARDIPINE
9	NIC001	EX	NIC001-011-009	1	IV NICARDIPINE
10	NIC001	EX	NIC001-011-010	1	PLACEBO

Obs	Start Date/Time of Treatment	End Date/Time of Treatment	Study Day of Start of Treatment	Study Day of End of Treatment
1	2007-10-12T23:25:00	2007-10-24T10:00:00	1	13
2	2007-10-14T16:30:00	2007-10-21T10:30:00	1	8
3	2007-11-10T18:00:00	2007-11-23T19:00:00	1	14
4	2007-12-02T06:10:00	2007-12-14T18:00:00	1	13
5	2007-12-08T10:15:00	2007-12-19T00:30:00	1	12
6	2007-12-15T21:00:00	2007-12-27T20:00:00	1	13
7	2007-12-30T16:15:00	2008-01-12T16:15:00	1	14
8	2008-01-09T11:35:00	2008-01-22T08:00:00	1	14
9	2008-02-24T21:30:00	2008-03-09T00:00:00	1	14
10	2008-03-01T21:00:00	2008-03-13T19:40:00	1	13

Sample ADaM Domains Metadata

Obs	Domain	DomainLabel	DomainKeys
1	ADSL	Subject Level Analysis Dataset	USUBJID, STUDYID

Obs	Purpose	RepeatingYN	Structure
1	Analysis Dataset	No	One record per patient

Sample ADaM ADSL Metadata

Obs	Domain	Variable Position	Variable Name	Variable Label	Variable Type	Variable Category	Variable Length
1	ADSL	1	STUDYID	Study Identifier	Char	Text	40
2	ADSL	2	USUBJID	Unique Subject Identifier	Char	Text	40
3	ADSL	3	SUBJID	Subject Identifier for the Study	Char	Text	40
4	ADSL	4	SITEID	Study Site Identifier	Char	Text	40
5	ADSL	5	RFSTDTC	Subject Reference Start Date/Time	Char	Text	64
6	ADSL	6	RFSTDT	Reference Start Date/Time (N)	Num	Datetime	8
7	ADSL	7	ENDTC	Subject Reference End Date/Time	Char	Text	64
8	ADSL	8	AGE	Age	Num	Integer	8
9	ADSL	9	AGEU	Age Units	Char	Text	10
10	ADSL	10	COUNTRY	Country	Char	Text	3
11	ADSL	11	SEX	Sex	Char	Text	2
12	ADSL	12	SEXN	Sex (N)	Num	Integer	8
13	ADSL	13	RACE	Race	Char	Text	80
14	ADSL	14	RACEN	Race (N)	Num	Integer	8
15	ADSL	15	ETHNIC	Ethnicity	Char	Text	40

Obs	SAS Format	Sort Order	Submit YN	Core	In All Data Sets YN	Definition
1		1	Yes	Req	No	= SDTM.DM.STUDYID
2		2	Yes	Req	Yes	= SDTM.DM.USUBJID
3			Yes	Perm	Yes	= SDTM.DM.SUBJID
4			Yes	Req	Yes	= SDTM.DM.SITEID
5			Yes	Req	Yes	= SDTM.DM.RFSTDTC
6	E8601DT.		Yes	Req	Yes	Convert RFSTDTC to SAS date
7			Yes	Req	Yes	= SDTM.DM.RFENDTC
8			Yes	Perm	Yes	= SDTM.DM.AGE
9			Yes	Perm	Yes	= SDTM.DM.AGEU
10			Yes	Perm	No	= SDTM.DM.COUNTRY
11			Yes	Perm	Yes	= SDTM.DM.SEX
12			Yes	Perm	Yes	=1 if male =2 if female
13			Yes	Perm	Yes	= SDTM.DM.RACE
14			Yes	Cond	No	=1 if "ASIAN" =2 if "BLACK OR AFRICAN AMERICAN" =3 if "WHITE" =4 if "OTHER"
15			Yes	Perm	No	=SDTM.DM.ETHNIC

Sample ADaM ADSL

Obs	Study Identifier	Unique Subject Identifier	Subject Identifier for the Study	Study Site Identifier	Date/Time of Birth (N)	Age	Age Units	Country	Sex	Sex (N)
1	NIC001	NIC001-011-001	001	011	1988-10-24	19	YEARS	USA	M	1
2	NIC001	NIC001-011-002	002	011	1928-07-20	80	YEARS	USA	M	1
3	NIC001	NIC001-011-003	003	011	1961-01-03	47	YEARS	USA	M	1
4	NIC001	NIC001-011-004	004	011	1935-09-18	73	YEARS	USA	F	2
5	NIC001	NIC001-011-005	005	011	1927-07-10	81	YEARS	USA	F	2
6	NIC001	NIC001-011-006	006	011	1954-08-11	54	YEARS	USA	F	2
7	NIC001	NIC001-011-007	007	011	1964-03-27	44	YEARS	USA	M	1
8	NIC001	NIC001-011-008	008	011	1951-11-28	57	YEARS	USA	F	2
9	NIC001	NIC001-011-009	009	011	1942-06-25	66	YEARS	USA	F	2
10	NIC001	NIC001-011-010	010	011	1963-04-06	45	YEARS	USA	F	2
11	NIC001	NIC001-011-011	011	011	1980-11-05	28	YEARS	USA	F	2
12	NIC001	NIC001-011-012	012	011	1974-10-07	34	YEARS	USA	F	2
13	NIC001	NIC001-011-013	013	011	1944-01-09	65	YEARS	USA	F	2
14	NIC001	NIC001-011-014	014	011	1961-02-15	48	YEARS	USA	M	1
15	NIC001	NIC001-011-015	015	011	1953-04-25	55	YEARS	USA	M	1

Obs	Race	Race (N)	Subject Reference Start Date/Time	Reference Start Date/Time (N)	Subject Reference End Date/Time
1	WHITE	3	2007-10-12T23:25:00	2007-10-12T23:25:00	2007-10-24T10:00:00
2	WHITE	3	2007-10-14T16:30:00	2007-10-14T16:30:00	2007-10-21T10:30:00
3	WHITE	3	2007-11-10T18:00:00	2007-11-10T18:00:00	2007-11-23T19:00:00
4	WHITE	3	2007-12-02T06:10:00	2007-12-02T06:10:00	2007-12-14T18:00:00
5	WHITE	3	2007-12-08T10:15:00	2007-12-08T10:15:00	2007-12-19T00:30:00
6	WHITE	3	2007-12-15T21:00:00	2007-12-15T21:00:00	2007-12-27T20:00:00
7	WHITE	3	2007-12-30T16:15:00	2007-12-30T16:15:00	2008-01-12T16:15:00
8	WHITE	3	2008-01-09T11:35:00	2008-01-09T11:35:00	2008-01-22T08:00:00
9	WHITE	3	2008-02-24T21:30:00	2008-02-24T21:30:00	2008-03-08T00:00:00
10	WHITE	3	2008-03-01T21:00:00	2008-03-01T21:00:00	2008-03-13T19:40:00
11	WHITE	3	2008-03-11T13:30:00	2008-03-11T13:30:00	2008-03-24T18:00:00
12	WHITE	3	2008-03-16T02:45:00	2008-03-16T02:45:00	2008-03-28T18:00:00
13	WHITE	3	2008-04-05T20:00:00	2008-04-05T20:00:00	2008-04-18T18:00:00
14	WHITE	3	2008-04-05T23:15:00	2008-04-05T23:15:00	2008-04-12T07:30:00
15	WHITE	3	2008-04-14T18:00:00	2008-04-14T18:00:00	2008-04-15T20:30:00

Sample ADaM ADSL Continued

Obs	Safety Population Flag	Safety Population Flag (N)	Full Analysis Set Population Flag	Full Analysis Set Population Flag (N)	Intent-To-Treat Analysis Set Population	Intent-To-Treat Analysis Set Population	Description of Planned Arm
1	N	0	N	0	Y	1	PLACEBO
2	Y	1	Y	1	Y	1	NICARDIPINE .15
3	Y	1	Y	1	Y	1	NICARDIPINE .15
4	N	0	N	0	Y	1	PLACEBO
5	N	0	N	0	Y	1	PLACEBO
6	N	0	N	0	Y	1	PLACEBO
7	Y	1	Y	1	Y	1	NICARDIPINE .15
8	Y	1	Y	1	Y	1	NICARDIPINE .15
9	Y	1	Y	1	Y	1	NICARDIPINE .15
10	N	0	N	0	Y	1	PLACEBO
11	Y	1	Y	1	Y	1	NICARDIPINE .15
12	N	0	N	0	Y	1	PLACEBO
13	N	0	N	0	Y	1	PLACEBO
14	N	0	N	0	Y	1	PLACEBO
15	Y	1	Y	1	Y	1	NICARDIPINE .15

Obs	Date of First Exposure to Treatment	Time of First Exposure to Treatment	Datetime of First Exposure to Treatment	Date of Last Exposure to Treatment	Time of Last Exposure to Treatment	Datetime of Last Exposure to Treatment
1	2007-10-12	23:25:00	2007-10-12T23:25:00	2007-10-24	10:00:00	2007-10-24T10:00:00
2	2007-10-14	16:30:00	2007-10-14T16:30:00	2007-10-21	10:30:00	2007-10-21T10:30:00
3	2007-11-10	18:00:00	2007-11-10T18:00:00	2007-11-23	19:00:00	2007-11-23T19:00:00
4	2007-12-02	06:10:00	2007-12-02T06:10:00	2007-12-14	18:00:00	2007-12-14T18:00:00
5	2007-12-08	10:15:00	2007-12-08T10:15:00	2007-12-19	00:30:00	2007-12-19T00:30:00
6	2007-12-15	21:00:00	2007-12-15T21:00:00	2007-12-27	20:00:00	2007-12-27T20:00:00
7	2007-12-30	16:15:00	2007-12-30T16:15:00	2008-01-12	16:15:00	2008-01-12T16:15:00
8	2008-01-09	11:35:00	2008-01-09T11:35:00	2008-01-22	08:00:00	2008-01-22T08:00:00
9	2008-02-24	21:30:00	2008-02-24T21:30:00	2008-03-08	00:00:00	2008-03-08T00:00:00
10	2008-03-01	21:00:00	2008-03-01T21:00:00	2008-03-13	19:40:00	2008-03-13T19:40:00
11	2008-03-11	13:30:00	2008-03-11T13:30:00	2008-03-24	18:00:00	2008-03-24T18:00:00
12	2008-03-16	02:45:00	2008-03-16T02:45:00	2008-03-28	18:00:00	2008-03-28T18:00:00
13	2008-04-05	20:00:00	2008-04-05T20:00:00	2008-04-18	18:00:00	2008-04-18T18:00:00
14	2008-04-05	23:15:00	2008-04-05T23:15:00	2008-04-12	07:30:00	2008-04-12T07:30:00
15	2008-04-14	18:00:00	2008-04-14T18:00:00	2008-04-15	20:30:00	2008-04-15T20:30:00

Sample ADaM ADSL CDI Program