PharmaSUG2010 - Paper CD08

The SE Domain as Treatment Data Store:
Using the SDTM Subject Elements (SE) domain to drive analyzing treatment

Jeremy Gratt, Modular Informatics, LLC, Thornwood, NY

ABSTRACT

One of the advantages of the adoption of CDISC SDTM and ADaM data standards is being able to leverage the domain structures to improve consistency and quality of clinical trial analysis. Treatment can present a barrier to reporting standardization and automation due to the wide range of treatment designs. This paper demonstrates how the SDTM SE trial design domain can be used as a central treatment data store and used to build treatment into the ADaM datasets and clinical trial reports. The SE domain is further leveraged with a sponsor-specific SUPPTE domain that is used to "tag" SE with additional treatment meta-data used to define the treatment analysis. The approach presented in this paper is based on an implementation with a biometrics programming group focused on Phase 1 studies with complex trial designs.

INTRODUCTION

When SDTM datasets are created, usually the SE trial design domain is created as an after-thought to provide additional treatment information about the study. SE contains a set of treatment intervals for each subject covering both active medication as well as off-treatment periods (like screening, washout and post-treatment). This paper will describe implementation details for using SE as the central treatment data store, and show how SE is used downstream in ADaM and report creation.

SE will be used to build treatment variables into ADaM datasets, and this treatment data is further used to generate reports with data-driven treatment column headers. An additional SUPPTE domain is created to attach additional treatment meta-data to SE, including an active medication flag, treatment display text for reports, and treatment display sort order.

Implementation of SE with the additional SUPPTE treatment meta-data domain provided the programming group with a means to centrally control treatment, and the flexibility needed to handle the reporting for complex Phase 1 crossover studies.

HANDLING SIMPLE TREATMENT DESIGNS

Figure 1. Epochs for a Parallel Study Design

Figure 1 displays the treatment epochs for a simple parallel design study. For a design like this, the following figure outlines a common method for handling treatment on the way to creating reports.

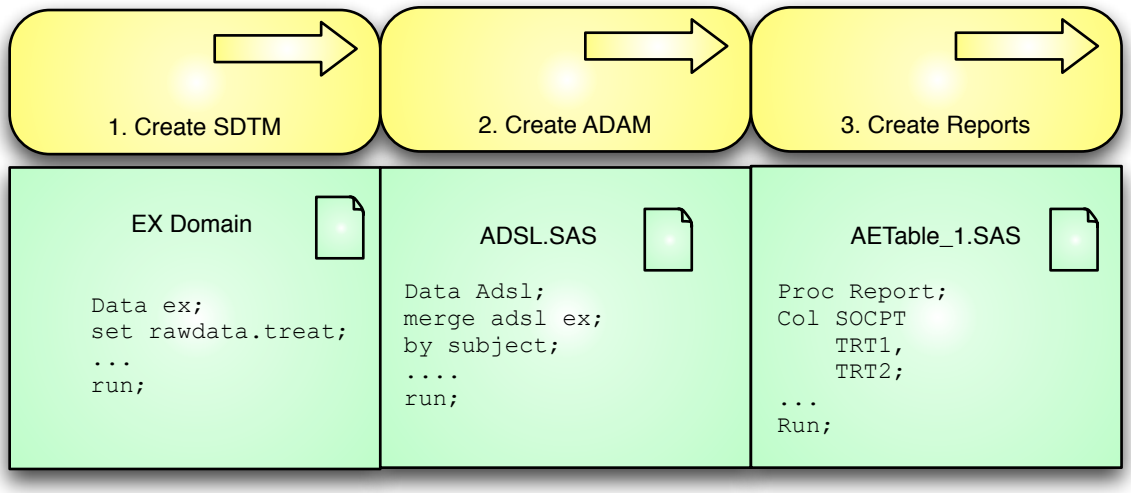


Figure 2. Report-Building Process for a Parallel Study Design

In this case, the treatment data is handled as follows:

1. The SDTM EX Domain contains the treatment exposure and dosing information from the CRF, and it is used as the source of actual treatment data.
2. The EX Domain is merged into ADSL to derive a unique treatment start date, stop date and actual treatment code for each subject. Other ADaM datasets read from ADSL.
3. In the reporting stage, treatment column headers are hard coded, or sometimes coded via a SAS format.

This is a fairly standard method for simple study designs. But the EX domain only describes the treatment dosing information, it does not give any description to the pre-drug, post-drug, or between-drug periods. Information about screening, run-in, wash-out, post-treatment, and post-study periods may be needed for proper analysis. For example, knowing when post-treatment ends and post-study begins may be important for safety analyses and this kind of information is not captured in the EX domain. For more complex designs, the SE domain was seen as a more appropriate choice because of it's completeness: it contains actual treatment information (active and non-active) for all time periods within the study.

Hard coding of treatment column headers presents a challenge because for each new study, the reporting programs must be customized to handle the change in treatment column headers. This decreases code reuse.

THE SE DOMAIN

For this example, we will assume the following treatment design:

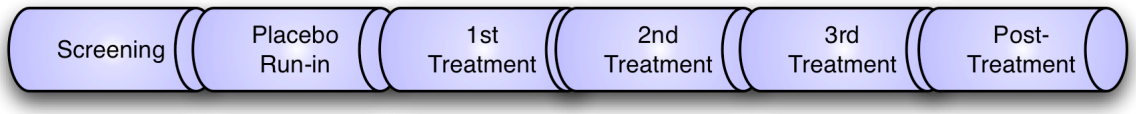


Figure 3. Epochs for a Crossover Design With Placebo Run-in

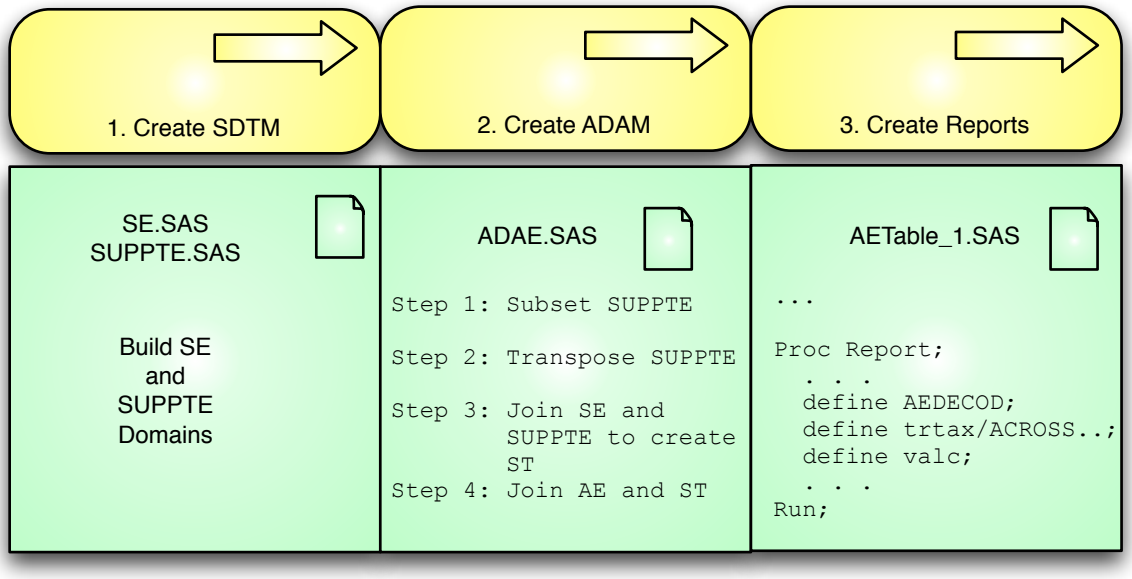
The SE Domain on the next page contains a set of non-overlapping treatment intervals spanning the entire time period that each subject was in the study. SE contains both active medications epochs (1st treatment, 2nd treatment, 3rd treatment) as well as off-treatment epochs (screening, placebo run-in, post-treatment). Note that in all sample datasets in this paper, extraneous variables are not displayed.

The SE Domain

USUBJID	ARM	SESTDTC	SEENDTC	ETCD	ELEMENT	EPOCH
101	ABC	2006-06-01	2006-06-09	SCRN	SCREEN	SCREENING
101	ABC	2006-06-09	2006-06-10	RIN	RUN-IN	PLACEBO RUN-IN
101	ABC	2006-06-10	2006-06-20	CA10	CURE-ALL 10MG	1st TREATMENT
101	ABC	2006-06-20	2006-06-24	CA15	CURE-ALL 15MG	2nd TREATMENT
101	ABC	2006-06-24	2006-06-30	FXL50	FIXALL 50MG	3rd TREATMENT
101	ABC	2006-06-30	2006-07-05	FUP	FOLLOW-UP	POST-TREATMENT
102	ACB	2006-06-01	2006-06-09	SCRN	SCREEN	SCREENING
102	ACB	2006-06-09	2006-06-10	RIN	RUN-IN	PLACEBO RUN-IN
102	ACB	2006-06-10	2006-06-20	CA10	CURE-ALL 10MG	1st TREATMENT
102	ACB	2006-06-20	2006-06-24	FXL50	FIXALL 50MG	2nd TREATMENT
102	ACB	2006-06-24	2006-06-30	CA15	CURE-ALL 15MG	3rd TREATMENT
102	ACB	2006-06-30	2006-07-05	FUP	FOLLOW-UP	POST-TREATMENT

OVERVIEW OF THE TREATMENT HANDLING PROCESS USING SE

The figure outlines the new treatment handling method that was used:



In this case, the treatment data is handled as follows:

1. The SE and SUPPTE domains are created.
2. A standardized, 4 step process is used to join treatment data with domain data (in this case, AE). Step 1 uses PROC SORT, Step 2 uses PROC TRANSPOSE, and Steps 3 and 4 use PROC SQL.
3. Tables are data-driven using the PROC REPORT ACROSS feature.

BUILDING TE, SUPPTE AND IDENTIFYING ACTIVE MEDICATIONS

BUILDING TE: The TE Domain contains the treatment codes stored in ETCD and corresponding ELEMENT decodes. It is used to build SE.

The TE Domain

ETCD	ELEMENT
SCRN	SCREEN
RIN	RUN-IN
CA10	CURE-ALL 10MG
CA15	CURE-ALL 15MG
FXL50	FIX-ALL 50MG
FUP	FOLLOW-UP

BUILDING SUPPTE: We create SUPPTE to contain additional treatment meta-data for selected treatment codes.

The SUPPTE Domain

IDVAR	IDVARVAL	QNAM	QLABEL	QVAL
ELEMENT	CURE-ALL 10MG	ANALLBL1	Analysis Label 1	Actual Treatment Analysis
ELEMENT	CURE-ALL 15MG	ANALLBL1	Analysis Label 1	Actual Treatment Analysis
ELEMENT	FIX-ALL 50MG	ANALLBL1	Analysis Label 1	Actual Treatment Analysis
ELEMENT	CURE-ALL 10MG	ACTMDFL1	Active Medication Flag 1	Y
ELEMENT	CURE-ALL 15MG	ACTMDFL1	Active Medication Flag 1	Y
ELEMENT	FIX-ALL 50MG	ACTMDFL1	Active Medication Flag 1	Y
ELEMENT	CURE-ALL 10MG	TRTAX1	TRT Label for Reports 1	Cure-All ~ 10mg
ELEMENT	CURE-ALL 15MG	TRTAX1	TRT Label for Reports 1	Cure-All ~ 15mg
ELEMENT	FIX-ALL 15MG	TRTAX1	TRT Label for Reports 1	Fix-All ~ 50mg
ELEMENT	CURE-ALL 10MG	TRTAN1	TRT Order for Reports 1	1
ELEMENT	CURE-ALL 15MG	TRTAN1	TRT Order for Reports 1	2
ELEMENT	FIX-ALL 15MG	TRTAN1	TRT Order for Reports 1	3
ELEMENT	RUN-IN	ANALLBL2	Analysis Label 2	Treatment Run-In Analysis
ELEMENT	RUN-IN	ACTMDFL2	Active Medication Flag 2	Y
ELEMENT	RUN-IN	TRTAX2	TRT Label for Reports 2	Placebo ~ Run-In
ELEMENT	RUN-IN	TRTAN2	TRT Order for Reports 2	1
ELEMENT	CURE-ALL 10MG	ANALLBL3	Analysis Label 3	Cure-All vs. Fix-All Analysis
ELEMENT	CURE-ALL 15MG	ANALLBL3	Analysis Label 3	Cure-All vs. Fix-All Analysis
ELEMENT	FIX-ALL 50MG	ANALLBL3	Analysis Label 3	Cure-All vs. Fix-All Analysis
ELEMENT	CURE-ALL 10MG	ACTMDFL3	Active Medication Flag 3	Y

ELEMENT	CURE-ALL 15MG	ACTMDFL3	Active Medication Flag 3	Y
ELEMENT	FIX-ALL 50MG	ACTMDFL3	Active Medication Flag 3	Y
ELEMENT	CURE-ALL 10MG	TRTAX3	TRT Label for Reports 3	Cure-All
ELEMENT	CURE-ALL 15MG	TRTAX3	TRT Label for Reports 3	Cure-All
ELEMENT	FIX-ALL 15MG	TRTAX3	TRT Label for Reports 3	Fix-All
ELEMENT	CURE-ALL 10MG	TRTAN3	TRT Order for Reports 3	1
ELEMENT	CURE-ALL 15MG	TRTAN3	TRT Order for Reports 3	1
ELEMENT	FIX-ALL 15MG	TRTAN3	TRT Order for Reports 3	2

In this example, we analyze treatment 3 different ways.

- Analysis 1 is an analysis of actual treatment, comparing Cure-All 10mg, Cure-All 15mg and Fix-All 50mg
- Analysis 2 is for analyzing the safety of the treatment run-in period.
- Analysis 3 is an analysis comparing all Cure-All titrations vs. Fix-All.

SUPPTE can be better visualized after it is merged with TE. For example, in the following table, TE is joined with SUPPTE for Analysis 1:

The TE Domain with SUPPTE Merged for Analysis 1 – Actual Treatment Analysis

ETCD	ELEMENT	ACTMDFL	ANALLBL	TRTAX	TRTAN
SCRN	SCREEN				
RIN	RUN-IN				
CA10	CURE-ALL 10MG	Y	Actual Treatment Analysis	Cure-All ~ 10mg	1
CA15	CURE-ALL 15MG	Y	Actual Treatment Analysis	Cure-All ~ 15mg	2
FXL50	FIX-ALL 50MG	Y	Actual Treatment Analysis	Fix-All ~ 50mg	3
FUP	FOLLOW-UP				

- ACTMDFL – Active Medication Flag
- ANALLBL – Analysis Label
- TRTAX – the actual text that will be displayed in treatment column headers. In this example, ~ is the split character used by PROC REPORT to split lines over several rows.
- TRTAN – The display order of the treatment column headers

CREATING ADAM DATASETS WITH SE AND SUPPTE

When creating ADaM Datasets, SE and SUPPTE are joined with the relevant domain data, subsetting to the specific analysis needed. This is a sample of the standard code that was used, in this case selecting Treatment Analysis 1. There are 4 steps to the algorithm:

```
/* Step 1: Extract treatment meta-data from SUPPTE for Treatment Analysis 1 */
Proc sort data = sdtm.suppte
  (where=(QNAM in ('ANALLBL1','ACTMDFL1','TRTAX1','TRTAN1') ))
  out = suppte;
  by studyid idvarval;
run;

/* Step 2: Transpose treatment meta-data to produce anallbl, actmdfl, trtax, trtan
vars */
proc transpose data = suppte
  out=suppte2 (drop=_:
               rename = ( anallbl1 = anallbl
                          actmdfl1 = actmdfl
                          trtax1   = trtax
                          trtan1   = trtan ));
  by studyid idvarval;
  var qval;
  id qnam;
run;

/* Step 3: Join SE with treatment meta-data vars from transposed SUPPTE */
proc sql;
  create table st (label= "Subject Treatment Dataset" where = (actmdfl = "Y"))
  as
  select a.usubjid,
         b.anallbl format = $200. Label = "Analysis Label",
         b.actmdfl format = $1. Label = "Active Medication Flag",
         a.etcd as trta format = $8. Label = "Actual Treatment",
         b.trtax format = $200. Label = "Actual Treatment Label for Display",
         b.trtan format =8. Label = "Actual Treatment Display Order",
         input(a.sestdtc,yymmdd10.) as SESTDT format=yymmdd10.,
         input(a.seendtc,yymmdd10.) as SEENDT format=yymmdd10.
  From sdtm.se as a left join suppte2 as b
  On a.element = b.idvarval
  Order by usubjid, sestdt;
  Quit;

/* Step 4: Join ST from Step 3 with domain data – in this case AE */
Proc sql;
  Create table ae2 as
  Select a.*, b.anallbl, b.trta, b.trtan, b.trtax, b.sestdt, b.seendt
  From ae1 as a left join st as b
  On a.usubjid=b.usubjid and sestdt le a.aestdt lt seendt
  Order by studyid, usubjid, aestdt;
Quit;
```

The first 3 steps create the intermediate ST (subject treatment) dataset, as seen on the next page. ST is simply SE with variables from SUPPTE, subset to those records in SE with ACTMDFL="Y".

ST: Steps 1 through 3 generate the ST Dataset created in ADaM Program:

USUBJID	ANALLBL	ACTMDFL	TRTA	TRTAX	TRTAN	SESTDT	SEENDT
101	Actual Treatment	Y	CA10	Cure-All ~ 10mg	1	2006-06-10	2006-06-20
101	Actual Treatment	Y	CA15	Cure-All ~ 15mg	2	2006-06-20	2006-06-24
101	Actual Treatment	Y	FXL50	Fix-All ~ 50mg	3	2006-06-24	2006-06-30
102	Actual Treatment	Y	CA10	Cure-All ~ 10mg	1	2006-06-10	2006-06-20
102	Actual Treatmnet	Y	FXL50	Fix-All ~ 50mg	3	2006-06-20	2006-06-24
102	Actual Treatment	Y	CA15	Cure-All ~ 15mg	2	2006-06-24	2006-06-30

In Step 4, AE is merged with ST that was just created previously

Step 4: Merge of ST with AE data using adverse events that fall between SESTDT and SEENDT.

USUBJID	AEDECOD	TRTA	TRTAX	TRTAN	SESTDT	AESTDT	SEENDT
101	DERMATITIS CONTACT	CA10	Cure-All ~ 10mg	1	2006-06-10	2006-06-15	2006-06-20
102	HEADACHE	FXL50	Fix-All ~ 50mg	3	2006-06-20	2006-06-22	2006-06-24

In this example, Subject 101 had a Dermatitis Contact adverse event that occurred while the Cure-All 10mg treatment was taken. The TRTAX and TRTAN variables included in the final ADAE ADaM dataset will be used to define treatment headers in the tables.

GENERATING REPORT OUTPUT WITH DATA-DRIVEN TREATMENTS

In the previous section, it was demonstrated how treatment variables, including TRTAX and TRTAN, were joined to AE data to produce ADAE. This section shows how these variables are used to create tables with data-driven treatment column headers.

This reporting method depends on the PROC REPORT ACROSS option. Please see the papers ' "ACROSS" in PROC REPORT ' By Jiang Jin from NESUG 16 and "Using ACROSS Variable in PROC REPORT PROCEDURE" by Min Fu from the NESUG'96 Proceedings for more information. The key to using PROC REPORT ACROSS in these cases is using a special, non-printing DUMMY numeric variable.

As handled in Jiang Jin's paper, the reporting program generates reporting data vertically as follows:

RPTDATA – Report data for creating an AE table by Preferred Term

AEDECOD	TRTA	TRTAX	TRTAN	VALC	DUMMY
DERMATITIS CONTACT	CA10	Cure-All ~ 10mg~(n=10)	1	3 (30%)	1
DERMATITIS CONTACT	CA15	Cure-All ~ 15mg~(n=10)	2	2 (20%)	1
DERMATITIS CONTACT	FXL50	Fix-All ~ 50mg~(n=10)	3	0	1
HEADACHE	CA10	Cure-All ~ 10mg~(n=10)	1	0	1
HEADACHE	CA15	Cure-All ~ 15mg~(n=10)	2	4 (40%)	1
HEADACHE	FXL50	Fix-All ~ 50mg~(n=10)	3	3 (30%)	1

Note that the reporting program appends the treatment population count (n=10) to the TRTAX variable.

This is the sample PROC REPORT call:

```
proc report data=rptdata nowd headline headskip split="~";
  col aedecod trtax, (valc) dummy;
  define aedecod / group order=data  "Preferred~Term" width=14 left;
  define trtax / across order=data "-Treatment Group- ~ " ;
  define valc / " " width=14;
  define dummy / noprint;
  break after aedecod / skip;
run;
```

And this is sample output using data-driven treatment column headers:

```
                       ------------Treatment Group--------------
                       Cure-All      Cure-All      Fix-All
Preferred              10mg          15mg          50mg
Term                   (n=10)        (n=10)        (n=10)
-----------------------------------------------------------------

Dermatitis Contact     3 (30%)       0             0

Headache               0             4 (40%)       3 (30%)
```

This demonstrates how the treatment display variable TRTAX defined in SUPPTE is now displayed in the output table.

CHOOSING DIFFERENT TREATMENT ANALYSES FROM SUPPTE

The SUPPTE domain defined in this paper had 3 different types of treatment analyses and the previoius examples showed how to generate a table using Analysis 1. Suppose now that a table using Analysis 3 was required. Analysis 3 compares all titrations of Cure-All combined vs. Fix-All. Treatment analysis 3 can be selected in the ADaM program by altering Steps 1 and 2 to extract the TRTAX3, TRTAN3 elements from SUPPTE:

```
/* Step 1: Extract treatment meta-data from SUPPTE for Treatment Analysis 3 */
Proc sort data = sdtm.suppte
  (where=(QNAM in ('ANALLBL3','ACTMDFL3','TRTAX3','TRTAN3') ))
  out = suppte;
  by studyid idvarval;
run;

/* Step 2: Transpose treatment meta-data to produce anallbl, actmdfl, trtax, trtan
vars */
proc transpose data = suppte
  out=suppte2 (drop=_:
               rename = ( anallbl3 = anallbl
                          actmdfl3 = actmdfl
                          trtax3   = trtax
                          trtan3   = trtan ));
  by studyid idvarval;
  var qval;
  id qnam;
run;
```

This small change in the ADaM program would produce the following report:

	--Treatment Group---	
Preferred Term	Cure-All (n=20)	Fix-All (n=10)
Dermatitis Contact	3 (30%)	0
Headache	4 (40%)	3 (30%)

CONCLUSION

Combining SE as a treatment data store with SUPPTE containing sponsor-defined treatment meta-data makes it is possible to standardize treatment handling code and create flexible reporting programs with data-driven treatment column headers.

The key to this approach is the definition of an active medication flag (ACTMDFL). Assignment of this flag to treatment codes in SE makes it possible to select the treatments to be used for analsyis, and to combine different actual treatments into new groups. This provides project programmers with a flexible method for handling and reporting on a wide variety of treatment designs.

ACKNOWLEDGEMENTS

Special Thanks to James Nezamis of Mannkind Corporation for his insights.

CONTACT INFORMATION

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

Name:	Jeremy Gratt
Enterprise:	Modular Informatics, LLC
Address:	73 Orchard Street
City, State ZIP:	Thornwood, NY 10594
E-mail:	jgratt@modularinformatics.com