This paper describes an automation system for producing a Case Report Form Tabulation (CRT) listing. For the purposes of this paper, the authors define a CRT listing as a clinical report that interleaves a series of individual data listings (e.g., demographic characteristics, laboratory data, and vitals) by patient.

The system approach used in this paper applies an efficient data integration and patient listing reporting process while providing a simple yet powerful programming interface to produce CRT listings.

The SAS product used in this paper is SAS BASE on a UNIX platform.

INTRODUCTION

SAS programmers in the pharmaceutical industry support the clinical data analysis and reporting process. One of the more frequently requested reports is the Case Report Form Tabulation (CRT). It is a clinical report that interleaves individual data listings (e.g. demographic characteristics, laboratory data, and vitals) together for each patient.

There are a number of methods to produce a CRT. A typical CRT listing program creates intermediate SAS datasets, and stores them in a temporary SAS library. Then the reporting routine reads in the temporary data to interleave each patient's information in the listings. An individual listing in the CRT is called a panel listing. The clinical study team decides how many panel listings will be included in a CRT. Figure 1 shows an example of Demographic Characteristics panel listing.

This paper presents an alternative approach for an automation system that is simpler, more efficient, and easier to use. The automation system is a SAS macro that consists of five modules: a hit list module, a driver module, a data module, a reporting module and a customized module. The programmer uses SAS macro parameters to select or drop the individual panel listings that are provided by the system, to be included or excluded in the CRT. The customized module is an optional module for user to include any other panel listings that are not provided from the system. The system flowchart is shown in Figure 2.

A hit list module identifies the subset of study population that will be included in the CRT listings. The main program controls execution of the individual macros, and populates the macro variables needed for each patient.

The system provides a core set of panel listings and an optional non-core set of panel listings. Each panel listing involves sub-routines in both data module and reporting module. The core set listings are: 1) Demographic Characteristic, 2) Patient Conclusion, 3) Adverse Experience, 4) Concomitant Medication, 5) Vital Signs, and 6) Study Medication. The system provides the flexibility that user can select to include all core set panel listings to the CRT or drop any one of panel listing from core set to the CRT. The non-core set of panel listings are a collection of panel listings from a template library. Currently, this paper includes several non-core panel listings. They are: 1) ECG Quantitative Data, 2) Cardiovascular History, 3) CHF Etiology, and 4) Presenting Conditions.

The customized module accepts user's panel listings that are not provided from the system. The user needs to prepare data and reporting sub-modules for the panel listings to be included in the CRT.

The automation approach creates a panel listing templates library including core and non-core panels for the programmer to include or exclude from the CRT. It is
an easier and efficient process and is discussed in the following sections using sample SAS code.

### DATA SOURCE

All data for the system are retrieved from the Global Data Mart (GDMart). It is a collection of SAS datasets in a standard structure that are used for reporting clinical data. The data mart building process entails mapping data fields from the Oracle database into SAS datasets based on the GDMart specification.

For illustrative purposes, nine data sets, one for each panel listing, from a hypothetical clinical trial are used in this paper. The variables in each data set are:

<table>
<thead>
<tr>
<th>Data File Name</th>
<th>Core Set</th>
<th>Variable List</th>
</tr>
</thead>
<tbody>
<tr>
<td>dem</td>
<td>Y</td>
<td>pid, dem_race, dem_racet, dem_sex, dem_age, dem_wt, dem_ht, dem_bsa</td>
</tr>
<tr>
<td>end</td>
<td>Y</td>
<td>pid, end_cmp, end_reas</td>
</tr>
<tr>
<td>pcmed</td>
<td>Y</td>
<td>pid, med_drug, med_gens, med_dose, med_frq, route, med_ind, med_rtmf, med_bdat, med_edat, dur, med_ctna</td>
</tr>
<tr>
<td>dose</td>
<td>Y</td>
<td>pid, visit, dos_adm, dos_frq, dos_lvl, dos_bdat, dos_rtmf, dos_edat</td>
</tr>
<tr>
<td>ae</td>
<td>Y</td>
<td>pid, ae_body, ae_pref, ae_bdat, ae_therp, ae_rtmf, ae_dur, ae_episd, ae_int, ae_ser, ae_act, ae_rel, ae_vtext, ae_etext</td>
</tr>
<tr>
<td>vital</td>
<td>Y</td>
<td>pid, visit, vit_val, vit_prmc, vit_rtmf</td>
</tr>
<tr>
<td>mhist</td>
<td>N</td>
<td>pid, mh_diag, mh_pref, mh_disc1, mh_pcm</td>
</tr>
<tr>
<td>cvhis</td>
<td>N</td>
<td>pid, cv_vis, cv_dat, cv_disc, cv_dis, cv_lpp, cv_ansyn</td>
</tr>
<tr>
<td>ecg</td>
<td>N</td>
<td>pid, vist, ecg_val, ecg_prmc, ecg_rtmf</td>
</tr>
<tr>
<td>chfcs</td>
<td>N</td>
<td>pid, cc_iscm, cc_cauc, cc_cau, cc_diagd, cc_ncp</td>
</tr>
</tbody>
</table>

### HIT LIST MODULE

A hit list dataset identifies the subset of patients that will be included in the CRT listings.

```sas
%macro hitlist;
  data hitlist;
    set mart.trx;
    if trx ^= ' ';
    label trx = 'TREATMENT GROUP';
    keep pid trx;
    proc sort; by pid;
  run;
%mend hitlist;
```

### DATA MODULE

The data are read from a standard data mart. The data mart build programs not only produce a standard data structure but also perform data derivation and data manipulation. It is the best source for all clinical reporting needs. The data module performs data derivation and manipulation when it is needed. The following sample codes are from the data module.

```sas
%macro crt_data;
%macro setdata(dsin=, dsout=);
  data &dsout;
    merge mart.&dsin(in=b) hitlist (in=a);
    by pid;
    if a and b;
    proc sort; by trxl pid;
  %mend setdata;
%if %upcase(&drop) ^= DEM %then %do;
  %setdata(dsin=dem, dsout=dem1);
%end;
%if %upcase(&drop) ^= END %then %do;
  %setdata(dsin=end, dsout=end1);
%end;
%if %upcase(&drop) ^= PCMED %then %do;
  %setdata(dsin=pcmed, dsout=pcmed1);
  /* more data step statements */
%end;
%mend crt_data;
```

### REPORTING MODULE

The primary reporting SAS macro, `crt_prn`, consists of two sections. The first internal SAS macro, `def`, defines all of the define statements that PROC REPORT needs. If the user does not create this macro, then PROC REPORT will apply default characteristics. The main body of `crt_prn` is used to select the panel listings, read the data and to display it with a DO loop for selecting one patient at a time.

A panel listing may contain one or several observations. The system may generate many thousands of pages of output having only a portion of each page filled with text. The reporting module uses `formdlim` option to put multiple outputs on one page.
CUSTOMIZED MODULE

The system is flexible that allows user to insert customized panel listings into CRT. The user needs to provide a pair of sub-modules for each customized panel listing, one for gathering the data and one for reporting the data. The following sample code is an example for a customized LABORATORY TEST DATA panel listing. The macros labd and lab are the data sub-module and reporting sub-module respectively.

%macro labd;
 *****************************************;
 *Read in MART data needed for the listing;
 *****************************************;
data dem0(keep=trxl pid age sex race);
 set mart.dem;
 length sex race $1. age $2.;
 age = compress(put(dm_age, 8.));
 sex = left(dm_sex);
 race = left(dm_racec);
run;

data lab1;
 merge mart.lab(in=a)
 dem0
 hitlist(in=b);
by pid;
if a and b;
length inv_rng $12 lab_prms $30;
inv_rng = trim(left(lab_rlo)) ||' - '||trim(left(lab_rhi));
lab_prms = trim(left(lab_prm));
if lab_rval = . then delete;
format lab_rval 8.1;
label trxl = 'TREATMENT GROUP'
lab_prm = 'LAB TEST'
sex = 'S^E^X'
race = 'R^A^C^E'
age = 'Age'
visit = 'OBSERVATION'
lab_rtmf= 'RELATIVE*^DAYS'
lab_rval= 'LAB ^VALUE'
%end;
%macro lab;
 *****************************************;
 *Write out PATIENT CONCLUSION DATA*;
 *****************************************;
data endl;
 run;

%end;
inv_rng = 'Investigator^Reference Range'
lab_nrfg = 'F^1'
lab_bfg = 'F^2'
lab_cfg = 'F^3'
lab_runt = 'LAB^UNIT'
lab_prms = 'LABORATORY PRAMETER';

run;
proc sort; by trxl pid;
run;
%mend labd;

%macro lab;

title7 " PATIENT NUMBER: &&p&jj TREATMENT GROUP: &&t&jj ";

proc report data=lab1 headline split='^'
   by trxl pid;
   column ( "=^LABORATORY TEST DATA^=
         age sex race lab_prm lab_rtmf visit
         lab_rval lab_nrfg lab_bfg lab_cfg inv_rng
         lab_runt
   
   where pid = "&&p&jj";
   define age/ order width=3;
   define sex/ order width=1;
   define race/ order width=1;
   define lab_prm/ order width=20;
   define lab_rtmf/ order width=20;
   break after lab_prm/skip;
run;
%mend lab;

DRIVER MODULE

The driver module controls execution of the individual macros, and populates the macro variables needed for each patient.

%macro crt ;
%crt_data;
*****************************************;
* Create the macro variables for each pid 
*****************************************;
%let pnum=0;
data _null_
   set _crt nobs=nobs;
call symput("p" || compress(put(_n_, 10.,)), pid);
call symput("t" || compress(put(_n_, 10.,)), &sortby);
call symput("pnum", put(_nobs, 4.));
run;

SYSTEM INVOCATION AND OUTPUT

The system is invoked by selecting the SAS macro parameters library, the hit list of patients required for the CRT, and the driver module executes macro calls.

These individual modules are simplified in this example for illustrative purposes. However, the authors developed the automation approach to be implemented within a typical complex SAS reporting environment.

The macro parameter descriptions are shown as follows:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Parameter Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sortby</td>
<td>A variable list to specify the sort order of CRT.</td>
</tr>
<tr>
<td>drop</td>
<td>Specify a panel listing from core set to be excluded in CRT</td>
</tr>
<tr>
<td>mhist</td>
<td>A non-core Presenting Condition panel listing to be included in CRT (Y/N)</td>
</tr>
<tr>
<td>chfcs</td>
<td>A non-core CHF Etiology panel listing to be included in CRT (Y/N)</td>
</tr>
<tr>
<td>ecg</td>
<td>A non-core ECG Quantitative panel listing to be included in CRT (Y/N)</td>
</tr>
<tr>
<td>cvdhis</td>
<td>A non-core Cardiovascular History panel listing to be included in CRT (Y/N)</td>
</tr>
<tr>
<td>insert1</td>
<td>User provides panel listing (with data and reporting sub-module) to be included in CRT</td>
</tr>
<tr>
<td>insert2</td>
<td>User provides panel listing (with data and reporting sub-module) to be included in CRT</td>
</tr>
</tbody>
</table>

The following codes show a sample system invocation.

%hitlist ;
%crt (sortby = pid ,
drop = vital,
ecg = y,
chfcs = y,
mhist = y,
cvdhis = ,
insert1 = lab,
insert2 = );
run;

The above invocation example will produce a CRT with following panel listings: 1) Demographic Characteristic, 2) Patient Conclusion, 3) Adverse Experience, 4) Concomitant Medication, 5) Study Medication, 6) ECG Quantitative Data, 7) CHF Etiology, 8) Presenting Experience, 4) Concomitant Medication, 5) Study Medication, 6) ECG Quantitative Data, 7) CHF Etiology, 8) Presenting Experience.
Condition, and 9) Laboratory Test Data. The output from the sample invocation are presented in Figures 3 through 5. Note that Figure 3 contains multiple panel listings on one page.

**CONCLUSIONS**

Using this new approach, we were able to produce CRT listings quickly and efficiently. This approach provides the following features:

* System is flexible and easy to use.
* Quick to set up; takes advantage of existing resources.
* Technically efficient;
  - short, simple source code; minimized validation and maintenance activity.
  - fast execution.
  - minimized output size.
* Provides a useful framework and mechanism for the rapid development and modification of an automated CRT system.

**REFERENCES**


**TRADEMARKS**

SAS is a registered trademarks of SAS, in the USA and other countries.

indicates USA registration.

**AUTHORS**

David C. Izard, M.S.
(610)917-6911(W)
E-mail: David.C.Izard@gsk.com

Shi-Tao Yeh, Ph. D
(610)917-5883(W)
E-mail: shi-tao_yeh-1@gsk.com