Customized Flowcharts Using SAS® Annotation
Abhinav Srivastva, PaxVax Inc., Redwood City, CA

ABSTRACT
Data visualization is becoming a trend in all sectors where critical business decisions or assessments are made. In pharmaceuticals, flowcharts are used to provide a high-level overview of the clinical trials data such as summarizing patients’ disposition, enrollments and summary of adverse events, medical history or dosing data. As opposed to most common layouts that SAS® can generate using newer SG procedures or GTL, these flowcharts have non-typical layouts and refer back to the concept of SAS/GRAPH Annotate Facility. The paper will have a brief discussion on some SAS annotation functions/macros paving the way for a SAS macro that allows users to make the flowcharts customizable to meet their requirements and enabling the production of several forms with minimal programming effort. This flexibility is achieved through a naming system that has been developed and is discussed in great detail in following sections.

INTRODUCTION
The flowcharts can take many forms and below is one example of Clinical Study status from Enrollment to Completion. It indicates the number of subjects who signed the Informed Consent form, followed by subjects who were randomized if not a screen failure. Based on Intent to Treat principle a Full Analysis Set (FAS) is defined further, leading to a summary of completions and discontinuations. This information can be represented in the form of a flowchart as in Figure 1 below –

![Figure 1: Clinical Study-level flowchart](image)

The macro discussed below allows user to make the flowchart customizable with the number of tiers or floors needed and establishing relevant linking between the individual elements. Some other sample layouts are displayed in Figure 2 below –

![Figure 2: Sample flowchart templates](image)
Customized Flowcharts Using SAS Annotation, continued

**SAS ANNOTATION FUNCTIONS**

The Annotate facility enables customization of the layout by drawing objects of interest through a set of annotation tools available in SAS. The annotation macros used for this flowchart are listed below in Table 1.

<table>
<thead>
<tr>
<th>Macros</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>%ARROW</td>
<td>%ARROW (x1, y1, x2, y2, color, line, size, angle, style)</td>
</tr>
<tr>
<td>%BAR</td>
<td>%BAR (x1, y1, x2, y2, color, line, style)</td>
</tr>
<tr>
<td>%LABEL</td>
<td>%LABEL (x, y, text-string, color, angle, rotate, size, style, position)</td>
</tr>
</tbody>
</table>

**Table 1: Annotation Macros**

Before invoking the macros the following needs to be submitted to compile the annotation macros –

%annomac

The complete list along with the syntax can be found by submitting a statement as-

%helpano
%helpano(label) /* for ‘Label’ macro and so on .. */

The key elements of annotation include:

- **FUNCTIONS** – label, bar, arrow
- **STYLING** – Color, Size, Style, Position
- **POSITIONING** – x, y, xsys, ysys

For more information on annotation please see reference [1] in the REFERENCE section.

**FLOWCHART MACRO NAMING SYSTEM**

The macro creates the flowchart taking into account the number of tiers or floors and the linking that is desired in the flowchart. These objects are marked below-

![Diagram](image)

**Figure 3: Naming System**

Each element in the flowchart is designated by an ID which is a combination of floor # and element #. Floors are counted from top to bottom and elements are counted from left to right. For the above example, the first element (top box) is
identified by ‘11’ (floor = 1, element = 1), the second element by ‘21’ (floor = 2, element = 1) and finally the bottom two elements will be assigned as ‘31’ & ‘32’ respectively. Assigning IDs this way allows for linking between them, for example ‘11 >21’ implies Box 1 on first tier (ID= 11) is linked with Box 1 on second tier (ID=21) and so on. These instructions can be summarized in a table as below:

<table>
<thead>
<tr>
<th>ID</th>
<th>Instructions</th>
<th>SAS Annotation Macros</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>Draw a box on Tier 1 as the first element</td>
<td>%BAR</td>
</tr>
<tr>
<td>21</td>
<td>Draw a box on Tier 2 as the first element</td>
<td>%BAR</td>
</tr>
<tr>
<td>31</td>
<td>Draw a box on Tier 3 as the first element</td>
<td>%BAR</td>
</tr>
<tr>
<td>32</td>
<td>Draw a box on Tier 3 as the second element</td>
<td>%BAR</td>
</tr>
<tr>
<td>11&gt;21</td>
<td>Draw a link connecting ID=11 and ID=21</td>
<td>%ARROW</td>
</tr>
<tr>
<td>21&gt;31</td>
<td>Draw a link connecting ID=21 and ID=31</td>
<td>%ARROW</td>
</tr>
<tr>
<td>21&gt;32</td>
<td>Draw a link connecting ID=21 and ID=32</td>
<td>%ARROW</td>
</tr>
</tbody>
</table>

Table 2: Tabular representation of Naming System

FLOWCHART MACRO

After having a grasp of the naming system in the previous section, let’s proceed with designing the flowchart. The flowchart macro is composed of 2 steps:

1. Create a comprehensive universal set of elements and links
2. Superimpose the desired flowchart onto the universal set in Step 1 to identify the layout of interest and hide extra objects from the universal set

In order to demonstrate the execution of the macro let’s consider the test case of Figure 1: Clinical Study-level Flowchart from the INTRODUCTION section.

STEP 1: CREATE A UNIVERSAL SET

Using the annotation functions listed earlier a universal set is created where several combinations of elements and links are drawn. The below figure is a 4x4 set (4 tiers and 4 elements/tier) combination but this can be expanded to as many tiers as needed depending on the page settings. Every object in the universal set has a unique ID as marked below per the naming system.
An annotated dataset is created using the annotation macros. A snippet of the annotation for ID=11 (leftmost box on tier 1) is shown in the below display.

<table>
<thead>
<tr>
<th>ID</th>
<th>LINK</th>
<th>WHEN</th>
<th>XSYS</th>
<th>YSYS</th>
<th>COLOR</th>
<th>X</th>
<th>Y</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>00</td>
<td>a</td>
<td>3</td>
<td>3</td>
<td>black</td>
<td>0</td>
<td>N</td>
<td>MOVE</td>
</tr>
<tr>
<td>11</td>
<td>00</td>
<td>e</td>
<td>3</td>
<td>3</td>
<td>black</td>
<td>20</td>
<td></td>
<td>BAR</td>
</tr>
<tr>
<td>11</td>
<td>11&gt;21</td>
<td>b</td>
<td>9</td>
<td>9</td>
<td>black</td>
<td>-10</td>
<td></td>
<td>MOVE</td>
</tr>
<tr>
<td>11</td>
<td>11&gt;21</td>
<td>b</td>
<td>9</td>
<td>9</td>
<td>black</td>
<td>0</td>
<td>-10</td>
<td>ARROW</td>
</tr>
<tr>
<td>11</td>
<td>11&gt;22</td>
<td>b</td>
<td>9</td>
<td>9</td>
<td>black</td>
<td>0</td>
<td></td>
<td>MOVE</td>
</tr>
<tr>
<td>11</td>
<td>11&gt;22</td>
<td>b</td>
<td>9</td>
<td>9</td>
<td>black</td>
<td>27</td>
<td>-10</td>
<td>ARROW</td>
</tr>
<tr>
<td>11</td>
<td>11&gt;12</td>
<td>b</td>
<td>9</td>
<td>9</td>
<td>black</td>
<td>-17</td>
<td>17</td>
<td>MOVE</td>
</tr>
<tr>
<td>11</td>
<td>11&gt;12</td>
<td>b</td>
<td>9</td>
<td>9</td>
<td>black</td>
<td>10</td>
<td></td>
<td>ARROW</td>
</tr>
</tbody>
</table>

Display 1: Annotation for ID=11

Two new variables are created in the annotated dataset ID and LINK. LINK refers to the linkage for the given ID.

1. A bar is drawn with ID=11 using %BAR function. LINK variable is set to ‘00’
2. An arrow is drawn using %ARROW function originating from ID=11 as ‘11>21’ which is defined in LINK variable
3. An arrow is drawn using %ARROW function originating from ID=11 as ‘11>22’ which is defined in LINK variable
4. An arrow is drawn using %ARROW function originating from ID=11 as ‘11>12’ which is defined in LINK variable

A careful observation of the universal set will reveal several embedded layouts that can be obtained from this set, adding to the flexibility of the flowchart.

**STEP 2: SUPERIMPOSE DESIRED LAYOUT ON UNIVERSAL SET**

In this step, the desired layout is superimposed on the universal set and extraneous elements and links are hidden. Graphically when the test case (Figure 1) is superimposed on the universal set, the outcome is displayed in Figure 5.
The universal set serves as a framework with multiple embedded schemes or layouts that can facilitate any type of layout as long as it's defined in the set. The universal set can further be customized to accommodate complex structures.

In the process when Figure 1 is superimposed on the universal set we obtained our desired layout and extraneous objects (elements/links) were hidden from display as seen in Figure 5: Superimposition. It is important to note the ID’s of the elements and links after the superimposition. It can be summarized in Table 3 as below:

<table>
<thead>
<tr>
<th>Tier</th>
<th>ID</th>
<th>Links ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12</td>
<td>12&gt;22</td>
</tr>
<tr>
<td>2</td>
<td>22</td>
<td>22&gt;32</td>
</tr>
<tr>
<td>3</td>
<td>32</td>
<td>32&gt;41 &amp; 32&gt;43</td>
</tr>
<tr>
<td>4</td>
<td>41</td>
<td>41 &amp; 43</td>
</tr>
</tbody>
</table>

**Table 3: Elements and Links ID After Superimposition**

The Elements and Links ID from Table 3 are then passed as input to the flowchart macro call. When there is a match between the IDs from the above table with the IDs from the universal set (Display 1) those objects are kept and the rest are hidden by changing the style attribute ‘color’ to ‘white’. This style attribute trick is the most convenient way to hide extraneous objects. The macro call for this scenario can be developed as below. With every layout the macro call will need to be modified to fit its unique requirements.

```
%flowchart(dsn = example, 1)
  floor= 12>var1 12>22 # 22>var2 22>32 # 32>var3 32>41 32>43 # 41>var4 # 43>var5);
```

Here,

1. **dsn** = Input dataset (example) containing variables with text to display in the flowchart.

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>TEXT TO DISPLAY</th>
</tr>
</thead>
<tbody>
<tr>
<td>var1</td>
<td>IC Signed*(N=100)</td>
</tr>
<tr>
<td>var2</td>
<td>Rand. Set*(N=50)</td>
</tr>
<tr>
<td>var3</td>
<td>FAS Set*(N=90)</td>
</tr>
<tr>
<td>var4</td>
<td>Completed*(N=80)</td>
</tr>
<tr>
<td>var5</td>
<td>Terminated*(N=10)</td>
</tr>
</tbody>
</table>

‘*’ is the split character used to span text into multiple lines

2. **Tier 1:** Element ID ‘12’ is combined with ‘var1’ to show the text “IC Signed (N=100)” so it’s represented as ‘12>var1’. A blank space is used as a delimiter between ‘12>var1’ and an outward link ‘12>22’ connecting ‘12’ to ‘22’.

3. **Tier 2:** Element ID ‘22’ is combined with ‘var2’ to show the corresponding text so it’s represented as ‘22>var2’. A blank space is used as a delimiter between ‘22>var2’ and an outward link ‘22>32’ connecting ‘22’ to ‘32’.

4. **Tier 3:** Element ID ‘32’ is combined with ‘var3’ to show the corresponding text so it’s represented as ‘32>var3’. A blank space is used as a delimiter between ‘32>var3’ and outward links ‘32>41’ and ‘32>43’ connecting to ID ‘41’ and ‘43’ respectively.

5. **Tier 4:** Element ID ‘41’ is combined with ‘var4’ to show the corresponding text so it’s represented as ‘41>var4’. It does not have any outward link associated with it.

6. **Tier 4:** Element ID ‘43’ is combined with ‘var5’ to show the corresponding text so it’s represented as ‘43>var5’. It does not have any outward link associated with it.

‘#’ is the delimiter between each of the definitions in [2] through [6].

Please refer to the APPENDIX for complete flowchart macro used to program Figure 1.
CONCLUSION

The methodology described here can avoid programming every potential scenario of the flowchart thus saving time and effort. The two step process described above of creating a universal set and superimposing desired layout onto this universal set is very straightforward and allows a good measure of flexibility to the users. Further enhancement can be made to tackle even complex layouts and varying the size of the annotated objects (bars/arrows etc) as needed.

REFERENCES


ACKNOWLEDGMENTS

I would like to thank Michelle Pallas (PaxVax, Inc.) and Trudi White (PaxVax, Inc.) for the review and feedback of the paper.

CONTACT INFORMATION

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

Name: Abhinav Srivastva
Enterprise: PaxVax, Inc.
E-mail: asrivastva@paxvax.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.
APPENDIX
/*==========================================================================*
* MACRO USAGE: */
* DSN       = Input dataset
* FLOOR     = List each element and its links separated by '#' delimiter
*==========================================================================*/
/* Input Dataset */
data example;
  length txt $ 100;
  var='var1'; txt='IC Signed*(N=100)' output;
  var='var2'; txt='Rand. Set*(N=90)' output;
  var='var3'; txt='FAS Set*(N=90)' output;
  var='var4'; txt='Completed*(N=80)' output;
  var='var5'; txt='Terminated*(N=10)' output;
run;
/* Invoke annotation macros */
%annomac;
%macro flowchart(dsn= , floor= );
/* Process the parameters passed in 'floor=' argument */
%let counter=1;
  %do %while (%scan(&floor,&counter.,%str(#)) ne %str( ));
    %let zz = %scan(&floor.,&counter.,%str(#));
    %let element = 1; * identify elements ;
    %let n = 1; * identify links ;
    %let counter = %eval(&counter.+1);
  %end
/* Merge with input dataset */
proc sql noprint;
create table all_dsn as
  select a.*, b.txt
  from all2
    left join example b on ID=scan(ELEMENT,1,'>')
     and LINK eq '00'
     and VAR eq 'var1'
  left join example a on ID=scan(ELEMENT,2,'>')
     and LINK eq '00'
     and VAR eq 'var2'
  left join example c on ID=scan(ELEMENT,3,'>')
     and LINK eq '00'
     and VAR eq 'var3'
  left join example d on ID=scan(ELEMENT,4,'>')
     and LINK eq '00'
     and VAR eq 'var4'
  left join example e on ID=scan(ELEMENT,5,'>')
     and LINK eq '00'
     and VAR eq 'var5'
run;
from all2 a
   left join &dsn. b
   on a.var=b.var
   order by ID, LINK
;
quit;

*>(&-------------
|--------------------
|---------------------------------------------------------------------*
|--------------------
|------------------
|STEP 1: Create Universal Flowchart Template ----------------
|------------------
|------------------------------------------------------------------------------*
|------------------------------
|* Standard Bar dimensions ;
\let b_x1=  0;
\let b_x2= 20;
\let b_y1= 95;
\let b_y2= 80;

|------------------------------
|/* Annotation dataset */
|data anno;
|length style id link $10;
%do j=1 %to 4; * 4 tiers or floors ;
  %do i=1 %to 3; * 3 elements per tier ;
    ID="&j.&i."; LINK='00';
    when='a'; * Bars will be drawn at the end of annotation ;
      %system(3,3); * absolute scale will be used for drawing bars ;
      %bar(&b_x1.,&b_y1.,&b_x2.,&b_y2.,black,0,e);
    when='b'; * other annotation pieces will be drawn before bars;
      LINK="&j.&i.>%eval(&j.+1)&i.";
        %system(9,9); * relative scale will be used for drawing arrows ;
        %arrow(-10,0,0,-10,black,1,1,90,filled);
      if &i. ^= 1 then do;
        LINK="&j.&i.>%eval(&j.+1)%eval(&i.-1)";
          %system(9,9);
          %arrow(0,+10,-27,-10,black,1,1,90,filled);
        LINK="&j.&i.>&j.>%eval(&i.-1)";
          %system(9,9);
          %arrow(+17,+18,-10,0,black,1,1,90,filled);
        LINK="&j.&i.>%eval(&j.+1)%eval(&i.+1)";
          %system(9,9);
          %arrow(+20,-8,+27,-10,black,1,1,90,filled);
      end;
    else do;
      LINK="&j.&i.>%eval(&j.+1)%eval(&i.+1)";
        %system(9,9);
        %arrow(0,+10,+27,-10,black,1,1,90,filled);
    end;
  end;
LINK="&j.&i.>%j.>%eval(&i.+1)";
%system(9,9);
Customized Flowcharts Using SAS Annotation, continued

\%arrow\{-17,+17,+10,0,black,1,1,90\,filled\};

/* Text annotation */
LINK='00'; TROW='1';
\%cnt2txt;
\%LABEL\{-20,+2,*,black,0,0,1,'Arial',5\}; * Will be used to display text line #1;
TROW='2';
\%system(3,9);
\%LABEL(\%eval(&b_x1.+10),-2,*,black,0,0,1,'Arial',5); * Will be used to display text line #2;
\%txt2cnt1;
TROW = ' ';

\%let b_x1 = \%eval(&b_x1. + 30);
\%let b_x2 = \%eval(&b_x2. + 30);
\%end;

\%let b_x1= 0;
\%let b_x2= 20;

\%let b_y1 = \%eval(&b_y1-25);
\%let b_y2 = \%eval(&b_y2-25);
\%end;
run;

/* Hide unnecessary links (if any) obtained from above do-loop processing */
data anno;
  set anno;
  if findc(link,'5')>0 then color='white'; * Hide 5th tier as its not needed here;
  if substr(link,lengthn(link))='4' then color='white'; * Hide 4th element in each tier;
run;

/*------------------------------------------*/
/*-- STEP 2: Perform Match with Universal Flowchart Template Created in Step 1 --*/
/*------------------------------------------*/
data final_anno (drop=txt);
length ID LINK var $10 txt text $100;
if _n_=1 then do;
  declare hash h(dataset:"all_dsn");
  h.defineKey("ID", "LINK");
  h.defineData("txt","var");
  call missing(txt,var);
  h.defineDone();
end;
set anno (drop=text);
if h.find()>0 then do; * Hide extra element/links in the Universal template;
  color='white';
end;
if txt ne ' ' then do;
  if TROW='1' then TEXT=scan(txt,1,'*'); * '* is the delimiter between lines of text;
  if TROW='2' then TEXT=scan(txt,2,'*');
end;
run;
Customized Flowcharts Using SAS Annotation, continued

/* Output to ODS RTF destination */
ods rtf file="flowchart.rtf";
  proc ganno anno=final_anno ;
  run ;
  quit ;
ods rtf close;
%mend;

/* Macro call */
%flowchart(dsn = example, floor= 12>var1 12>22 # 22>var2 22>32 # 32>var3 32>41 32>43 #
  41>var4 # 43>var5);