

Developing Graphical Standards: A Collaborative, Cross-Functional Approach

Mayur Uttarwar, Seattle Genetics, Inc., Bothell, WA

Murali Kanakenahalli, Seattle Genetics, Inc., Bothell, WA

ABSTRACT

“A picture is worth a thousand words:” a quote proven by the fact that graphs are increasingly used in reporting clinical trial data. Conventionally, SAS Graph[®] procedures in conjunction with annotation datasets have been used to create graphs, but recently SAS Graph[®] Template Language (GTL) is gaining ground. GTL is very powerful in its ability to allow the building of complex and intricate graphs using a structured approach. However, the key factor in leveraging the full potential of GTL and maximizing program efficiency is to establish graphical standards. This paper addresses the need for graphical standards, showcases a collaborative approach with cross-functional partners to identify & standardize common graph attributes, and provides one style template & one graph template that can support multiple graph types.

KEYWORDS

Graph Template Language (GTL), Graphical Standards, Style Template, Graph Template.

INTRODUCTION

Graphs play a major role in data analysis, and it can be challenging to achieve a consistently high level of quality across numerous graphs produced in support of multiple presentations and submissions of clinical trial data. We witnessed that, due to a lack of graphical standards, there was an excessive waste of valuable resources, like time and manpower, put forth by the affected teams within our organization. To resolve the issue, a committee involving key stakeholders was created with the purpose of investigating the issue and finding a solution. Through multiple discussions and testing various options, it was determined that developing viable standards for the graphical outputs could potentially streamline the process and reduce resource consumption. The purpose of this paper is to illustrate the process of how the standards were developed in a collaborative and cross functional approach, and show one style template and one graph template that can be used to support multiple graph types.

WHY WE NEED TO STANDARDIZE?

Graphs are an integral part of data visualization and are complicated to create based on the amount and nature of the data to be presented. Despite the efforts of the programmers to create the best possible graphs, there is usually a need for extensive further editing of the graphs. The team that assembles the final programming outputs, including the graphs, and prepares the final submission bears the brunt of this process. The editing process for the graphics editors is particularly laborious since they need to edit each graph individually using external software, which results in the editing process being quite time consuming. It is beneficial to come up with a set of graphical standards that works in the best interest of the medical writing team, the programming team and other end users of the graphical outputs.

SELECTION OF STAKEHOLDERS

Graphs are used mainly for publications, presentations, regulatory submissions and clinical study reports in pharmaceutical companies. Our goal was to involve key stakeholders from cross-functional team to create a set of graphical standards that would work for all parties. The key stake holders include clinical programmers, medical writers and statisticians. The clinical programming team initiated and led this project; they play a crucial role via programming the plots and implementing the standards. Statisticians were included in the project team as they lead design of the graphs. Medical writers were involved as they are the main recipients of the graphs, and technical editing of the graphs is performed within that group.

APPROACH AND PROCESS

The teams came up with a set of standard attributes that all graphs were expected to have. It is noteworthy to say that not all graphs can have all identical attributes, and hence the common ground was identified and some attributes such as colors, line styles, symbols, and fonts were thought to be appropriate to standardize across multiple graphs. It was also expected that, if needed, the graphs should be easily editable, and thus all graphs should be created as “vector” plots. SAS Graphic Template Language (GTL) is highly flexible and easy to implement, and thus was selected as the standard approach to creating graphs. There are two components of GTL; the style template and the graph template. In this paper, we are emphasizing the style template that we attempted to use as a basis for standardizing the graphs along with the graph template used to create a variety of graphs.

It is often useful to create graphs with specific visual characteristics that do not have to be hard coded into the GTL for every graph created. For example, the settings for the following graph features can be easily modified:

- Font or font sizes
- Line or marker properties
- Colors

Because the default properties of nearly all GTL appearance-related options are obtained from the existing style, we attempted to customize the style template in such a way that the required look and feel could be enforced on multiple graphs; however, we did not want to eliminate the user interaction while creating a standard style template.

To fully customize the graph in terms of the subgroups, headers, footers, axis labels etc., that possesses the standard attributes, we expected that the user provides the number of subgroups that would be presented on the graph. In practice, there could be a significant variety of attributes that can be standardized across multiple graphs, but for the sake of simplicity, let’s assume that any 4 different types are required for the lines, marker symbols, and colors of the lines, fill colors and contrast colors. The standard graphs are expected to have all subgroups represented within these 4 different types of attributes. There could be N number of subgroups and, keeping this in mind, we had to develop a strategy to cycle through the limited number of attributes for all the subgroups in order to produce a standard graph. A style template was created using a SAS macro for which the user provides the total number of subgroups. In the code displayed below, the macro parameter “subgroups” captures the total number of subgroups desired by the user and any four different styles, marker symbols, line/fill colors and contrast colors are assigned to the macro variables “line”, “marker”, “color” and “contrastcol” respectively.

```
%do i=1 %to &subgroups.;

  style GraphData&i. from GraphData&i./
  lineStyle = %scan(&line.,%sysfunc(mod(%sysfunc(ceil(%sysevalf(&i./4))),4)))
  markerSymbol = %scan(&marker.,%sysevalf(%sysfunc(mod(%sysevalf(&i.-1),4))+1))
  color = %scan(&color.,%sysevalf(%sysfunc(mod(%sysevalf(&i.-1),4))+1))
  contrastColor =%scan(&contrastcol.,%sysevalf(%sysfunc(mod(%sysevalf(&i.-1),4))+1));

%end;
```

This partial style template was then used to create a scatter plot and a line plot. The plots are based on simulated data and are created for the sole purpose of demonstration. The partial graph templates for the given two plots are given as:

Scatter plot:

```
layout overlay /yaxisopts=(* Standard attributes for the Y-Axis *)
xaxisopts=(* Standard attributes for the X-Axis *)
cycleattrs=true;
scatterplot x=x_variable y=y_variable /group=group_variable name='Legend_Name' ;
```

Line plot:

```
layout overlay /yaxisopts=(* Standard attributes for the Y-Axis *)
xaxisopts=(* Standard attributes for the X-Axis *);
seriesplot x=x_variable y=y_variable /group=group_variable name='Legend_Name' ;
```

From the scatter plot shown in figure 1 (a), it can be seen that the four types of marker symbols with four different colors are produced in ascending order of the group numbers. Since there are no lines involved in the scatter plot, specifying the line type in the style template did not make any difference. A user input in terms of axes labels and the labels for the legend was incorporated while developing these graphs. In the case of the line plot in figure 1 (b), the marker symbols represent the data points and these points are connected by the lines. Since the line type for the first subgroups is 1 (i.e. a regular line), the differing attributes among the subgroups are the marker symbols and the colors.

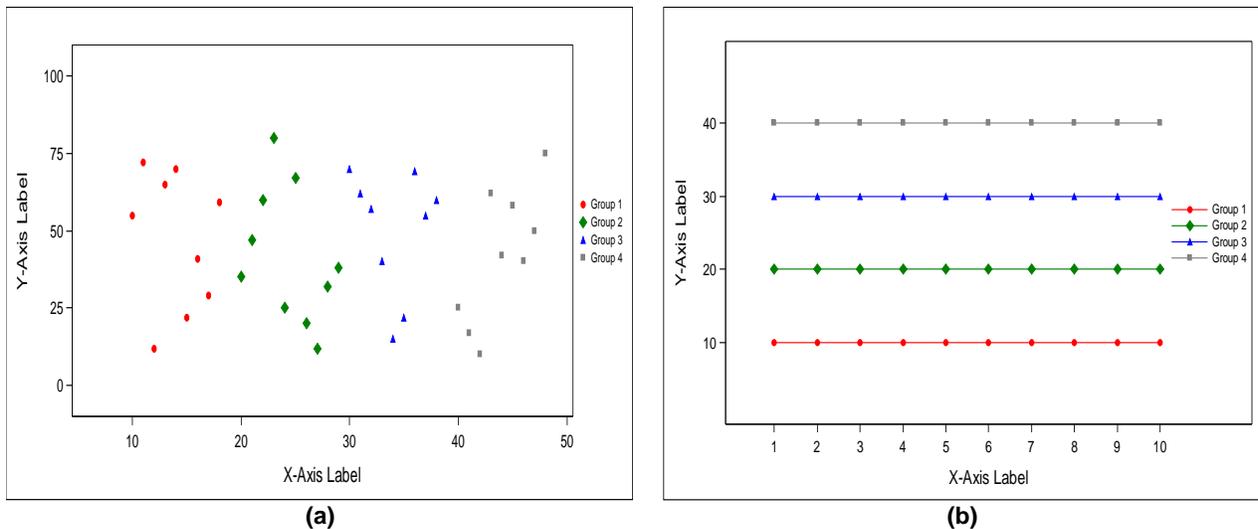


Figure 1. (a) Scatter Plot and (b) Line Plot with Standard Graphics Attributes

Applying the graphics standards to produce a survival plot (e.g. Kaplan-Meier curve) for four subgroups required a slightly different approach, since the plot involves a combination of a step plot and a scatter plot. The lines or steps represent the probability of survival for different subgroups and the marker symbols represent the censored observations. It can be noted that the marker symbols don't represent the interpolation points for the line in this case, but rather are overlaid on the line; hence we had to update the style template to accommodate this change. The rest of the contents of the template remained the same. The marker symbols are considered as a different set of subgroups; in this sense, each line plot has its own marker symbol for the censored observations. The Kaplan-Meier curves are shown in figure 2.

Style Template:

```
%do i=1 %to &subgroups.;

style GraphData&i. from GraphData&i./
lineStyle = %scan(&line.,%sysfunc(mod(%sysfunc(ceil(%sysevalf(&i./4)),4)))
color = %scan(&color.,%sysevalf(%sysfunc(mod(%sysevalf(&i.-1),4))+1))
contrastColor =%scan(&contrastcol.,%sysevalf(%sysfunc(mod(%sysevalf(&i.-1),4))+1));

style GraphData%sysevalf(&i.+&subgroups.) from GraphData%sysevalf(&i.+&subgroups.)/
markerSymbol = %scan(&marker.,%sysevalf(%sysfunc(mod(%sysevalf(&i.-1),4))+1))
contrastColor =%scan(&contrastcol.,%sysevalf(%sysfunc(mod(%sysevalf(&i.-1),4))+1));

%end;
```

Graph Template:

```
layout overlay / yaxisopts=(/* Standard attributes for the Y-Axis */)
xaxisopts=(/* Standard attributes for the X-Axis */);
stepplot x=x_variable1 y=y_variable1 / group=group_variable1 name='Legend_Name1';
scatterplot x=x_variable2 y=y_variable2/ group=group_variable2 name='Legend_Name2';
```

Based on figure 1 and figure 2, it was observed that a single style template could not be applied to assign the standard attributes to all the graphs, but it is anticipated that a major portion of graphs could be covered. Through GTL it was easy to create the majority of graphs in vector format (.emf format) so that the graphs could be edited in other software products like Microsoft PowerPoint if further enhancements were needed.

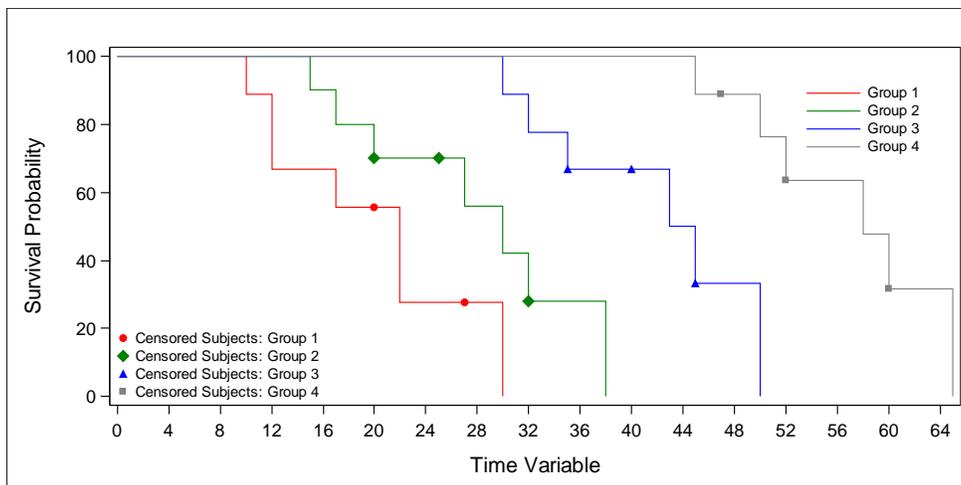


Figure 2. Kaplan-Meier Curves with Standard Graphics Attributes

ISSUES AND LEARNINGS:

While trying to standardize the graphs, we faced a few issues which are explained below:

Creating Border around pattern bars:

SAS lacks standard options within GTL for creating a border around pattern filled bars. To overcome this issue we created a standalone bar chart with a border and overlaid the one with patterns. Example code to achieve this end result is displayed below.

```

barchartparm y=_y x=_x /barwidth=0.5 index=_idx group=_grp name='trt'
fillattrs=(color=white) outlineattrs=(color=black);

barchartparm y=_y x=_x / display=(fillpattern) barwidth=0.5 index=_idx group=_grp skin=modern
name='trt' FILLPATTERNATTRS=(COLOR=black );
    
```

Graph size versus font size:

SAS GTL allows creating graphs based on templates. There is a correlation between the fonts within the graph and the size of the graph. Standardizing the graph size and font size was important to make graphs appear on one page and to make the legend appear within the allocated graph size. For a pre-defined page size, we used a trial and error method to optimize the design width and height, as well as the appropriate font sizes.

Vector graphics format and transparency:

Vector based graphics are more malleable than raster images. Thus, they are much more versatile, flexible and easy to use and edit. The most obvious advantage of vector images over raster graphics is they are quickly and perfectly scalable. Some medical writers prefer graphs to be delivered in vector graphics format and ODS destinations such as RTF or PDF. EMF image format is the only vector graphics format supported by ODS RTF. However, there is a drawback; GTL cannot apply transparency to vector graphics in EMF image format. SAS automatically creates the graph

in PNG raster format when transparency is applied. In our process we opted not to use transparency in the graphs we create.

Using same “graphdata” statements for overlay plots:

Kaplan Meyer plots usually have scatter plots to represent the censored observations and step plots to represent the probabilities of survival for multiple subgroups. Usually marker symbols and line style options can be defined within a single “graphdata” statement. When we used the same “graphdata” statement defining the line styles and marker symbols, the presentation of the symbols in the graph didn’t follow the options as defined in the “graphdata” statement. We decided to define the marker symbols and line styles in separate “graphdata” statements.

IMPACT OF CHANGES

Even though the implementation of graphical standards has streamlined the process of developing a variety of graphs, there are some challenges that need to be overcome. The impact on the clinical programming team is quite visible. The number of requests to modify graphs has been reduced as programmers now use the standard style template. Study programmers are expected to understand the templates and customize them per the study requirement or submission needs, and as a result enhance the applicability of the templates. In this process the efficiency of the cross functional teams increased as a result of a minimal amount of time spent on enhancing the graphs and putting less effort into bringing all the variety of graphs from all different studies under one standard format.

FUTURE SCOPE

Going forward, we want to identify the common statements within the graph template and style template for different graphs. We can minimize the work of the programmers by putting the common statements under globally defined macros. Instead of writing a whole graph template, we can call a stored macro. In this way it will be easy to bring standards and uniformity.

CONCLUSION

As graphs continue to play a major role in the analyses supporting clinical trials, there is a need to streamline the process of creating graphs within an organization so that programming and other relevant cross functional teams can spend minimal resources editing graphs or enhancing the visual attributes of the graphs. We have shown that a cross functional team establishing standards that align with the stakeholders’ interests can reduce the programming and medical writing resources needed to create commonly used graphs.

ACKNOWLEDGMENTS

We would like to thank Rajeev Karanam and Gordon Fancher for actively participating, supporting and guiding us throughout the process of developing the graphical standards. We received helpful comments from them, while this paper was written and we appreciate their crucial suggestions.

RECOMMENDED READING

- SAS/GRAPH® 9.2 Graph Template Language User’s Guide

CONTACT INFORMATION

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

Mayur Uttarwar
Seattle Genetics, Inc.
21823 - 30th Drive S.E.
Bothell, WA 98021
425-527-2284
muttarwar@seagen.com

Murali Kanakenahalli
Seattle Genetics, Inc.
21823 - 30th Drive S.E.
Bothell, WA 98021
425-527-2842
mkanakenahalli@seagen.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.