ABSTRACT

The project management-at-a-glance concept was developed to provide a communication graphic which could visibly provide the managers of several departments with an overview of all active projects. All the major milestones are predefined for simplicity. The detail between the milestones was also eliminated for clarity. The chart was designed to show multiple projects in a compressed format. Eight projects and 12 milestones can be shown on each page of the chart.

The Base SAS® package was utilized as the programming tool. The SAS/FSEDIT® product was used for data entry. The data for the report was stored in an IBM TSO sequential flat file. SAS/GRAPH® annotate tools are used to create the layered project management chart. The SAS system was chosen because of its strong ability to handle dates and custom generated graphics.

INTRODUCTION

The project management tool was designed to track mold die design for plastic injection molded parts. The mold die project management report was developed to furnish a simple tool which could provide visual information on multiple projects. The chart was designed to show multiple projects in a compressed format. Eight mold die projects and 12 milestones can be shown on each page of the chart.

The group which was to utilize this tool had never used project management concepts before so the system had to be as easy to use as possible. The users had previous experience with IBM's TSO product. Personal computer based project management tools were not considered because the users did not have a PC. The mainframe project management tools available were not used due to the time it would take to learn the systems.

The data for the report is stored in an IBM TSO sequential flat file. A SAS/FSEDIT® product is used for data entry. The SAS® System was chosen because of its strong ability to handle dates and custom generated graphics.

PROJECT MANAGEMENT-AT-A-GLANCE

The project management-at-a-glance concept was developed to provide a communication graphic which could visibly provide the managers of several departments with an overview of all active projects. All the major milestones of the projects are predefined for simplicity. The detail between the milestones are eliminated for clarity.

The format of the graphic is designed for readability at a glance. The page size is 8 1/2" by 11" for inclusion into monthly reports. Color is used for additional readability. The font sizes used for describing the milestone on the chart are large. The reports are printed in large quantities (30 to 40 copies) on four-color dot matrix printers.

Features of the chart:
1. eight projects can be viewed at one time,
2. twelve milestones are shown,
3. original specified project dates are shown,
4. current specified dates are included for each milestone,
5. seven weeks past the current date are tracked,
6. thirty seven weeks into the future are shown,
7. three lines of remarks on each project are available,
8. graph dates are updated based on the computer's current internal date,
9. numbered milestone descriptions are included on each page,
10. data entry is simplified by using a well-formatted screen with tabs, and
11. data entry is based on one screen for each milestone.

Color is used in several places on the graph as highlights. The major vertical and horizontal lines are blue. The titles, project labels, remarks, and the dates are also blue. The dividing line between history and forecast is red for its entire length the divide to graph into two parts. The descriptions of the milestones on the bottom of the page are green to help differentiate them from the rest of the page. The four color codes of the milestones on the body of the graph are:
- BLACK (used for the original milestone date)
- BLUE (used for on schedule current dates)
- RED (used for overdue milestones)
- GREEN (used for completed milestones)

LAYERING OF THE PROJECT MANAGEMENT GRAPHIC

The project management-at-a-glance chart is made up of several layers. The base layer remains the same for every run of the data. The next layer includes the dates which make up the heading of the graph. The third and final layer is the data itself.
THE BASE LAYER

The base layer is made up of vertical and horizontal lines. The lines are laid out in a grid pattern. The grid size chosen for this graph is 100 Y by 130 X. The computer draws the lines in the same manner that lines are drawn with a pencil. The lines are drawn by specifying the location to move the "pencil" to and then drawing the line by describing the first point and then the last point. The spacing is achieved by indexing the "pencil" and starting another line.

The initial X and Y location estimates were found by using graph paper. The locations were then transferred into a data file. Several trial runs using the computer graphics screen were made to make sure that the line locations were entered correctly before printing a trial grid.

Included in the first layer is all the default text contained on the graphic. The text is located on the paper by specifying color, justification, and location. The milestone descriptions are also part of the base layer.

THE DATE LAYER

The next layer is the date layer which runs across the entire top of the graphic. The graph is designed to have a minimum of maintenance, so the data file called WEEK is re-created each time the program is run. A DO loop is set up to create the forty-five weeks worth of dates. The internal current date from the computer is used to set up several variables.

The first step is to find the minimum date value by subtracting 7 weeks from the internal current date. The minimum date value is used as the starting point of the loop. The rest of the dates are found by adding 7 on each pass of the loop until 45 weeks (45 times through the loop) have passed. The loop also sets up the X and Y locations on the graph. The Y coordinate stays constant while the X location indexes by 3.

THE PROJECT LAYER

The third and most important layer is the data from the projects. Since there is one line of data for each milestone it may take as many as sixteen lines of data to make only one line on the graph. The start of the process involves sorting the data first by project and then by milestone. Several counters are created to aid in placing the project data in the correct X and Y location on the graph. A counter is also utilized to break the projects into several pages of eight projects per page.

Each milestone has an original and a current date. The program offsets the current and original dates by their Y location. The dates are given an X location in the first pass of the data. An algorithm based on the internal current date calculates where on the graph the milestone numbers should be placed.

Subroutines are used to calculate the Y location based on the projects counter (one through eight) and then whether the milestone is an original date or current date. The Y locations of the original and the current dates are offset by 15 spaces.

THE PROGRAMMING INVOLVED

The remainder of this discussion will continue with the programming specifics including coding examples of how the layering features are created. Several advanced topics of the SAS language family are used in the creation of the project management-at-a-glance graphic. Advanced topics include the following concepts:

1. using the SAS MACRO language,
2. the ANNOTATE function of PROC G Extraction, and
3. PUT statements of the Base SAS product.

Programming features:

1. setting up a FSEDIT data entry screen,
2. creating a grid for graphics,
3. algorithms which convert days between dates to X locations,
4. indexing of projects utilizing MACRO's,
5. creating a series of dates with annotate information using a DO loop, and
6. layered graphics.

SETTING UP THE FSEDIT DATA ENTRY SCREEN

The data entry screen is shown below. The information is broken into the overall project information and the individual milestone information.

![Figure 3](image-url)
CREATING A GRID FOR GRAPHICS

The graph is divided into a 110 Y by 130 X grid by simply specifying the use of cells, the HPOS, and the VPOS numbers in the GOPTIONS statement. A sample of the grid locations is shown in Figure 4.

GOPTIONS DEVICE=GDDM87 GUNIT=CELL HPOS=130 VPOS=110;

100 X 100 Y

<table>
<thead>
<tr>
<th>HISTORY</th>
<th>FORWARD</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROJECT</td>
<td>THREE</td>
</tr>
<tr>
<td>PROJECT</td>
<td>THREE</td>
</tr>
<tr>
<td>PROJECT</td>
<td>THREE</td>
</tr>
<tr>
<td>PROJECT</td>
<td>THREE</td>
</tr>
<tr>
<td>PROJECT</td>
<td>THREE</td>
</tr>
<tr>
<td>PROJECT</td>
<td>THREE</td>
</tr>
<tr>
<td>PROJECT</td>
<td>THREE</td>
</tr>
<tr>
<td>PROJECT</td>
<td>THREE</td>
</tr>
<tr>
<td>PROJECT</td>
<td>THREE</td>
</tr>
<tr>
<td>PROJECT</td>
<td>THREE</td>
</tr>
</tbody>
</table>

FIGURE 4.

ALGORITHMS WHICH CONVERT DATES TO X LOCATIONS

The current date is brought into the program using the TODAY function.

TODAY=TODAY();

The Minimum date and the maximum date boundaries are calculated at the time the program is run by going back 8 weeks and forward 37 weeks for a total of 45 weeks or 315 days.

MIN=TODAY-56;
MAX=TODAY+259;

The next step in the program is to convert the dates into X coordinates. Two equations translate the date into the X value for the original project finish date and the current project finish date. The equation for the original date and the current date are essentially the same.

Explanation of equation:

\[ X = \text{INT}\left(\frac{1}{3.5}\left((\text{ORIGINAL}-\text{TODAY})+56\right)+20\right); \]

STEP 1 Find the number of days between dates by subtracting the variable ORIGINAL from TODAY (the value will be between -56 and 259).

STEP 2 Convert the date difference to an X location by adding up the total days

56 + 259 = 315 days covered by the chart

INDEXING OF PROJECTS UTILIZING MACROS

SAS MACRO's are part of the base product of SAS. MACRO's can be used in a similar manner as a FORTRAN subroutine. The only difference is the MACRO's are placed at the top of the programmed code. The project management at a glance has several MACRO's. Using MACRO's results in a tremendous savings in programming code generation.

The data from the first data set is sorted by part number and milestone. The second data step sets up a counter, N, based on the first occurrence of a mold die project. The rest of the data also carries the same counter. Eight mold die projects can be shown per page of output.

The data is organized into groups of eight projects and placed into separate data sets to provide a manner for pagination. The M and N counters set up the position within the group of eight projects and which page the graph is to be printed on.

MACRO's are given names. The MACRO entitled GRAPH sets up the headers and places them in groups. In this MACRO the Y location is the important number. The Y axis is indexed by 15 spaces for each project. A consistent format is used for each data point. The format consists of the x,y locations, the color, style, weight, size, angle, and position of the text. Each description is saved as a point on the graph to be used in the ANNOTATE section of the program.

CREATING THE PROJECT LABEL

PROGRAM STATEMENTS (COMMENTS )

IF FIRST.PARTNUM THEN DO;
"USING THE FIRST PASS ONLY"
FUNCTION=LABEL;
"SPECIFY THE LABEL CATEGORY"
X=-2," x IS THE SAME EACH PASS"
Y=89-((M-1)*10);
"CALCULATE Y BASED ON THE M COUNTER"
COLOR=BLUE ";" SET COLOR TO BLUE"
STYLE=TRIPLEX ";" SPECIFY FONT STYLE"
LINE=1 ";" SPECIFY LINE WEIGHT"
TEXT=PARTNUM ";" VARIABLE NAME FOR PROJECT NAME"
SIZE=2," Font size"
ANGLE=0," Angle of Lettering"
POSITION=0," Relative Position of Letters"
OUTPUT &DATAOUT;
"SEND THE OUTPUT TO THE MACRO VARIABLE"
MEND;" END OF MACRO
The ORIGINAL and CURRENT MACRO's are also fired by the GRAPH MACRO. The two MACRO's position the milestone number on the project graph in the proper Y location. The X location is decided during the first data step. The Y location depends on which group the project fell into. The variable M controls the location of each project.

```
%MACRO CURRENT;
  * CREATE A MACRO CALLED CURRENT
  IF CURRENT GT 20 THEN STOP;
  * CHECK THE X VALUE OF THE VARIABLE CURRENT TO MAKE SURE IT FALLS ON THE CHART
  FUNCTION='LABEL';
  * SPECIFY THE LABEL CATEGORY
  X=XC+1 ;
  * INDEX THE X VARIABLE BY ONE FOR POSITIONING
  Y=88-((M-1)*10);
  * CALCULATE Y BASED ON THE M COUNTER
%MEND;' END OF MACRO

%MACRO ORIGINAL;
  * CREATE A MACRO CALLED ORIGINAL
  IF ORIGINAL GT 20 THEN STOP;
  * CHECK THE X VALUE OF THE VARIABLE CURRENT TO MAKE SURE IT FALLS ON THE CHART
  FUNCTION='LABEL ';
  * SPECIFY THE LABEL CATEGORY
  X=X0+1 ;
  * INDEX THE X VARIABLE BY ONE FOR POSITIONING
  Y= 88-((M-1)*10);
  * CALCULATE Y BASED ON THE M COUNTER AND MOVE 10 POSITIONS HIGHER
%MEND;' END OF MACRO
```

The MACRO ELIM is used to get rid of any data that is older than the seven week window.

```
%MACRO ELIM;
  * CREATE A MACRO CALLED ORIGINAL
  IF X LT 20 THEN DELETE ;
  * DELETE OLD DATA
  ELSE OUTPUT &DATAOUT; USE EVERYTHING ELSE
%MEND;' END OF MACRO
```

A flag is set to test if the milestone date is not complete and not greater than two weeks overdue. The flag is utilized in the chart to turn the milestone color red or green.

```
DIFF=TODAY-CURRENT;
  * CREATE VARIABLE WHICH IS TODAY'S DATE MINUS THE MILESTONES CURRENT COMPLETION DATE
  IF DIFF GE 14 AND COMPLETE NE 'C' THEN BS='Y';
  * IF BOTH THE MILESTONE COMPLETION DATE IS GREATER THAN TWO WEEKS AND THE PROJECT IS NOT COMPLETE THEN SET THE FLAG TO YES ELSE BS='N'; IF NOT SET THE FLAG TO NO IF BS='Y' THEN COLOR='RED' ;
  * IF THE FLAG IS YES SET THE COLOR TO RED IF COMPLETE='C' THEN COLOR='GREEN ';
  * IF THE FLAG IS NO SET THE COLOR TO GREEN
```

Creating a series of dates using a DO loop

The data step label DATA4 is the section that provides the dates across the top of the graph.

```
DATA4;
  TITLE 'WEEK';
  FOOTNOTE; CLEAR ALL FOOTNOTES
```

This graph is designed to have a minimum of maintenance, so the database called WEEK is recreated each time the program is run. A DO loop is set up to create the forty five weeks worth of dates. The variable TMN is used to count by 7 from the MIN date (TODAY - 56 days) to the maximum date.

```
DO i=1 TO 45 B
  * INITIATE THE LOOP
  TMN+7;
  * TAKE THE STARTING DATE, ADD A WEEK
  TEXT=TMN; CREATE THE VARIABLE TEXT
  FORMAT TEXT MMDDYY.;
  * FORMAT THE VARIABLE TEXT
  FUNCTION='LABEL '
  * SPECIFY THE LABEL CATEGORY
  X=INT((13.5)*(TEXT-TODAY)+56)+20)-1;
  * USE THE DATE CONVERSION EQUATION
  Y=90; Y IS CONSTANT
  ANGLE=90; TURN THE TEXT 90 DEGREES
END;' END THE LOOP
```

The dates are converted from numeric to character by putting the data set and then rereading the data set in the next data step. This step eliminates the error message from placing a number in a text field.

Layered graphics

The last DATA step sets or concatenates the three SAS System data sets into one data set called ALL. The data set ALL is annotated using PROC GSLIDE. The data that is concatenated was carefully formatted to be identical in all three data steps.

```
DATA ALL;SET DATA10 DATA6 DATAAO;
  JOIN THE THREE SAS DATASETS
  XSYS='S';YSYS='6';
  * SPECIFY
  TITLE1 H=4 F=TRIPLEX C=BLUE 'MOLD DIE DEVELOPMENT';
  * FIRST TITLE
  TITLE2 H=3 F=TRIPLEX C=BLUE 'PROJECT STATUS';
  * SECOND TITLE
  FOOTNOTE; CLEAR ALL FOOTNOTES
  PROC GSLIDE ANNOTATE=ALL;
  * CREATE THE SLIDE WITH ALL DATA JOINED TOGETHER
```

638
FUTURE DIRECTIONS
WITH CUSTOM GENERATED GRAPHICS

The project management-at-a-glance graphic and data entry screens are now tied together with an IBM TSO command language list (CLIST). An ISPF menu screen allows the end user to select from several options. The menu screen shields the end user from having to know the SAS System or very little else about TSO.

---IE REPORT AND GRAPHICS SUBMISSION SCREEN---
TIME: 13:12

COMMAND
---
1. INPUT - DATA ENTRY INPUT FOR MOLD DIE PROJECTS
2. GRAPH - RUN GRAPHIC OUTPUT OF PROJECTS
3. EDIT - EDIT DIRECTLY THE LISTING
PF3 - EXIT

ENTER SELECTION
---

FIGURE 6.

The most important lesson from this exercise is not the graphic project management tool because many others have created systems that are much more complete. The lesson is that a programming tool, the SAS System, can be manipulated to create custom reports with graphics. The graphic change in a fashion dictated only by the imagination of the programmer.

Another example of a custom generated graphic is a printed circuit board line balancing program which develops a manual assembly aid by entering the bill of material into an edit screen. The result is completely different from a project management system. The underlying principles, the MACRO's and the layering of the graphics, are identical. The ANNOTATE concepts can also be used with the standard SAS/GRAPH® PROC'S like GPLOT. Special information such as a specific note or calculation can be placed on the plot.

SAS, SAS/GRAPH, SAS/FEDIT are all registered trademarks of the SAS INSTITUTE, Cary, North Carolina.