

Developing a Dashboard to Aid in Effective Project Management

M. Paige Borden, University of Central Florida, Orlando, FL

Maureen Murray, University of Central Florida, Orlando, FL

Ali Yorkos, University of Central Florida, Orlando, FL

ABSTRACT

Accurately tracking programming support activities as a team manager can be a complex task. The overall coordination of the programming resources, testing support and production moves was hampered by the crude reporting functionality available to the change management system. This management challenge was solved by the development of the CMS IR Technical Support Dashboard utilizing SAS® Enterprise Guide and SAS/GRAPH functionality. Working together with the management team, the dashboard was created to provide quick access in an easy to read graphical display of numerous key performance indicators. There is also functionality providing multiple alerts as defined by the management team (e.g. an application that stays in “user test” for more than 72 hours). The CMS IR Technical Support Dashboard has provided an improved management tool to effectively and efficiently coordinate the multiple projects and constituencies. This presentation will detail the development, testing and release of the dashboard, while providing coding details and end-user requirements.

WHO WE ARE

Mission

The mission of the Office of Institutional Research is to provide information of the highest quality and which is both timely and easily accessible, and to facilitate and enhance decision-making, strategic planning, and assessment at the University of Central Florida.

Purpose

Institutional Research (IR) provides electronic and web-based dissemination of official information to the University community (including the Board of Trustees, the various colleges, departments and other academic and administrative units), external agencies, and the Florida Board of Governors (BOG). IR generates, supervises or develops all official University data reports and state-required reports, as well as providing end-user data solutions and training to facilitate decision-making processes. The director and staff serve on numerous university-wide committees and workgroups and assist with the collection and interpretation of institutional data, assist in planning academic programs, and participate in the implementation of evaluative procedures. The functions of the office support the entire university enterprise.

Background

Institutional Research has, as one of its primary objectives, the task of developing and implementing a university data warehouse. This enterprise-wide data system will facilitate the creation, access, and dissemination, by internal and external stakeholders, of institutional knowledge pertinent to the university. In collaboration with the data warehouse project stakeholder group, which represents a cross-section of key data users, the IR staff has established a data source of twelve years' worth of student reporting data. This warehouse of information serves as the foundation for the development of a wide variety of reporting applications using SAS Business Intelligence software.

INTRODUCTION

The university developed a change management system to coordinate our ERP implementation and maintenance. Within the past year, Institutional Research was granted access to the system to track the implementation and maintenance of our own reporting files, data warehouse development and on-line transactional web systems. Although this access greatly improved our tracking capabilities for our local project manager, the ability to create useful reports was not included in the system functionality. Dashboard functionality was evaluated and implemented to provide a better management tool for the overall project management.

The production version of the CMS IR Technical Support Dashboard was still under development at the time of this paper. Also, the initial design utilizes SAS Enterprise Guide, SAS/GRAPH and PROC GREPLAY functionality and was completed prior to the release of the SAS BI Dashboard 3.1 software. The authors intend to include the final output and a demonstration of the new SAS BI Dashboard 3.1 design at the conference.

FUNCTIONAL REQUIREMENTS

Overview of Functionality

Functional team members identified the data tables and data values that would need to be included to support the dashboard. In addition, several key performance indicators (KPI) were identified as items that needed detailed views to assist with project management. A mix of dials, graphs and/or gauges was desired, along with specific positive and negative alerts to identify specific successes or challenges. Additional requirements included a daily refresh and drill-down options for certain views.

The functional specifications identified 6 project team areas (Institutional Research, Data Warehouse, Reporting Database Service, University Analysis and Planning Support, Operational Excellence and Assessment Support and SACS) to include for the dashboard. These are the core project areas managed by IR personnel. Additionally, rules were specified regarding the amount of data to include on the dashboard using calendar guidelines for each status value (i.e. Development, User Test, Completed). A suggestion was made to the technical staff to limit the data selected for the dashboard rather than use the entire data set from the original tables. The limits were to be based on the project team areas and status fields.

The functional specifications also provided guidelines for field descriptions, such as displaying a 4-letter code versus providing a full name for developers.

Requested KPI Details

Multiple KPIs were identified as crucial views to manage the project teams. The method of providing the information – whether dial, graph or gauge – was left up to the technical team. KPIs included a) number of jobs by status, b) number of jobs by area, c) number of jobs by area by status, and d) number of jobs by developer. Some of the indicators (a and b) are aggregate graphs, whereas the KPIs included in both c) and d) are multiple graphs, providing details for each status or developer.

Requested Alerts

Both positive and negative alerts were identified as valuable reviews to quickly identify the successes and challenges of the current project. Positive alerts include e) completion status exceeds more than 25% of total jobs during the last 7 days, and f) no jobs in user test status older than 7 days. Negative alerts include g) jobs in user test older than 7 days, h) jobs in waiting for input status for more than 7 days, and i) more than 25% of jobs are still in assigned to developer status.

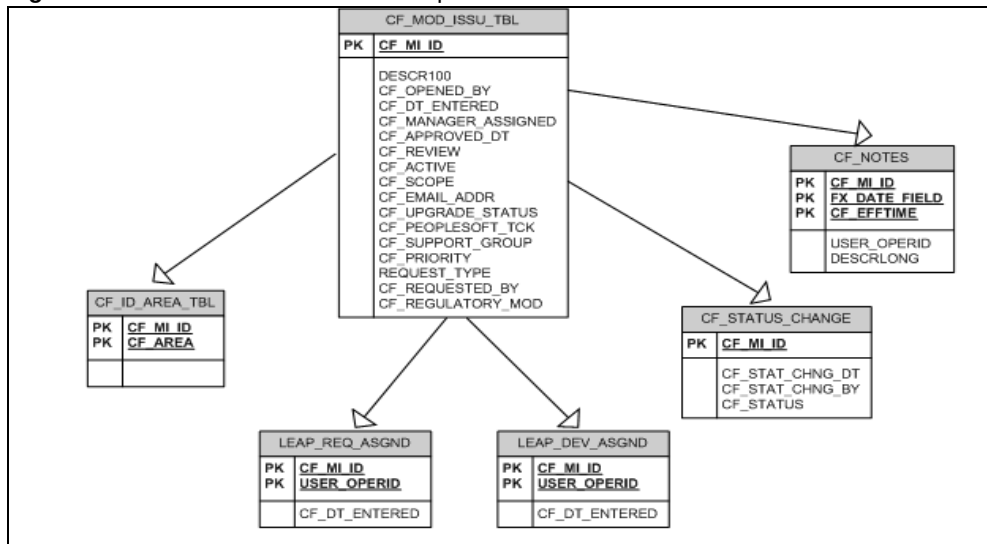
TECHNICAL DEVELOPMENT

Overview

After receiving the functional specifications, two programmers began the development process. Although working independently, the steps of development were the same. The first step in the technical development was to identify the data source and develop the underlying database tables that were needed for the dashboard implementation. Once the database tables were determined, it was essential to carefully examine the structure of the table fields that were required to produce the KPI details and the calculated alerts. Upon creating all the intermediate data sets, multiple graphs were produced that displayed KPI details, as well as text alerts.

Dashboard Table Development and Structure

The first step to creating the dashboard table involved the development of a single de-normalized dataset that included all the required fields. There were a minimum of 5 tables that were accessed to create this dashboard. The source table structures and the relationships are shown in Figure 1.

Figure 1: Table structure and relationships

The second step was to create 5 different datasets from the single de-normalized dataset to feed the dashboard graphs (Table 1).

Table 1: Dashboard graphs and its associated datasets

SAS Graph	Chart Type	Dataset
Jobs by Status	Vertical bar	MODS_STATUS
Jobs by Area	Vertical bar	MODS_AREA
Jobs by Developer	Horizontal bar	MODS_DEVELOPER
Jobs by Status by Area	Pie charts	MODS_AREA_STATUS
User Test Status	Pie chart	MODS_USER_TEST

Computed Fields for Alerts

The functional specifications called for text-based "Alerts" for specific areas. Table 2 summarizes the computed alerts detailing the associated fields, formulas and format types.

Table 2: Computed fields

Computed Alert	Formula	Type
Completed status of all jobs	CF_STATUS(COMP) / CF_STATUS(TOTAL)	Percentage
Assigned to developer of all jobs	CF_STATUS(ASGN) / CF_STATUS(TOTAL)	Percentage
User test status older than 7 days	CF_STATUS(UTST) and CF_STAT_CHNG_DT < DATETIME()-7	Sum
Wait for input status older than 7 days	CF_STATUS(WAIT) and CF_STAT_CHNG_DT < DATETIME()-7	Sum

The following code creates an annotation that displays the "Completed status of all jobs" alert as designed by Programmer 1. If the percentage is less than 25%, the text alert is displayed in red; otherwise the text is displayed in green.

```

data data1_anno3;
  set work.MODS_STATUS;
  where cf_status="COMP";
  length function $8 color $12 style $20 text $50 title $50 styl $20;
  hsys='3'; when='a';
  function='label';
  
```

```

style='swissb';
size=5;
xsys='3'; x=-57;
ysys='3'; y=95;
position='6';

    if percentage > .25 then color='cx006633';
    else color='cxcc0033';

    if color='cx006633' then
        text="Completed Status>25% of all jobs";
    else
        text="Completed Status<25% of all jobs";
    output;

style='swissb';
size=6;
xsys='3'; x=-35;
ysys='3'; y=108;
position='6';
text="ALERTS!";
color='cxcc0033';
output;
run;

```

SAS/GRAPH Development

The graphs for this dashboard were created in SAS Enterprise Guide 4. In cases where EG4 was not capable of executing certain functionalities, the graph code was altered in order to take full advantage of the capabilities of the SAS code. Vertical and horizontal bar charts, as well as pie charts, are used to display information for this dashboard. Since the dashboard output file is a .gif file, the developer needs to change the graph format from ActiveX to .gif during the initial graph development phase. This can be done by navigating to Graph menu item under Tools→Options→Results in EG4. Failure to make this alteration will cause alignment issues on the resulting dashboard page since the output styles for ActiveX are different than the .gif output styles. The annotation functionality was used to display text alerts and was integrated with the graph code. An example of a graph code that was used with two annotations (as developed by Programmer 1) is listed below:

```

PROC GCHART DATA=WORK.MODS_STATUS anno=data1_anno3
;
    VBAR    CF_STATUS /
    SUMVAR=COUNT
FRAME      TYPE=SUM
    OUTSIDE=sum
    noframe
    COUTLINE=BLACK
    width=7
    RAXIS=AXIS1
    MAXIS=AXIS2
    GAXIS=AXIS3
    GAXIS=AXIS3
    name="plot1"
PATTERNID=MIDPOINT

```

```
anno=data1_anno2;
RUN; QUIT;
```

PROC GREPLAY Procedure

The last step in graph development was to display all graphs in one dashboard page by using PROC GREPLAY procedure. The code used by Programmer 1 is detailed below.

```
PROC GREPLAY tc=tempcat nofs igout=work.gseg;
/* Define the areas of a custom dashboard template. */
tdef ucf des='UCF'

/* First n-chart row (various metrics) */
0/l1x = 0    l1y = 0
  ulx = 0    uly =100
  urx =100   ury =100
  lrx =100   lry = 0
1/l1x =30    l1y = 60
  ulx =30    uly = 95
  urx =80    ury = 95
  lrx =80    lry = 60
2/l1x =60    l1y = 60
  ulx =60    uly = 95
  urx =105   ury = 95
  lrx =105   lry = 60
3/l1x =0     l1y = 15
  ulx =0     uly = 75
  urx =60    ury =75
  lrx =60    lry =15
4/l1x =60    l1y = 0
  ulx =60    uly = 60
  urx =100   ury =60
  lrx =100   lry =0
5/l1x =0     l1y =0
  ulx =0     uly =40
  urx =45    ury =40
  lrx =45    lry =0
6/l1x =1     l1y =60
  ulx =1     uly =85
  urx =31    ury =85
  lrx =31    lry =60
;

/* Replay the individual indicators into the appropriate areas in the custom
  dashboard template. */
template = ucf;
treplay
  0:titles 1:plot1 2:plot2 3:plot3 4:plot4 5:plot5 6:PL0T6;
run;
quit;
```

Dashboard Output Designs

The two competing dashboard designs were very similar in output. Programmer 1 created the output in Figure 2, while Programmer 2 created the Figure 3 output.

Figure 2: Programmer 1 dashboard display

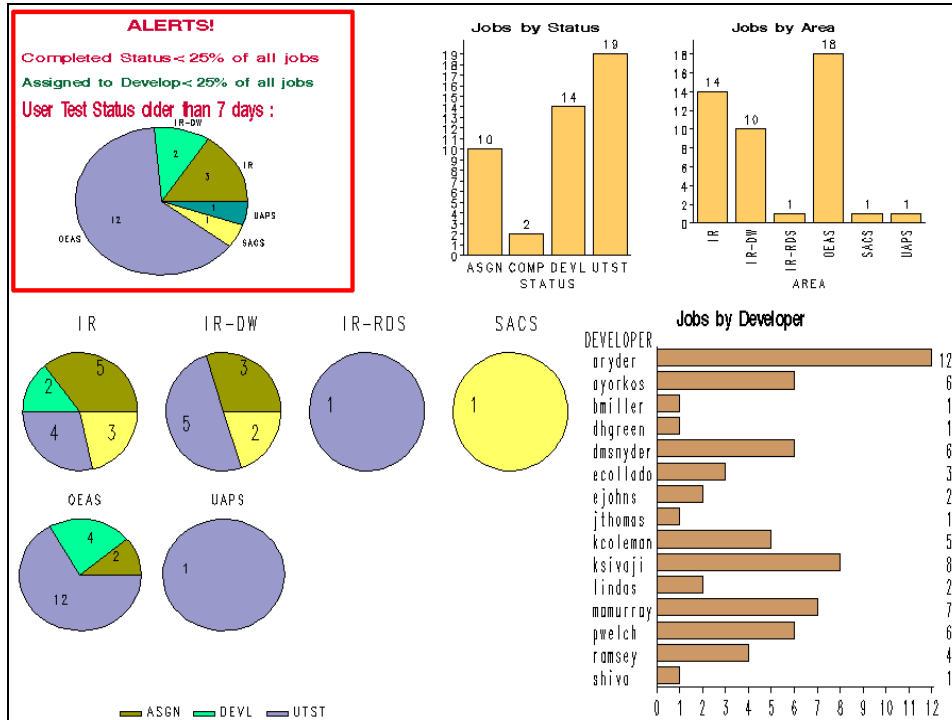
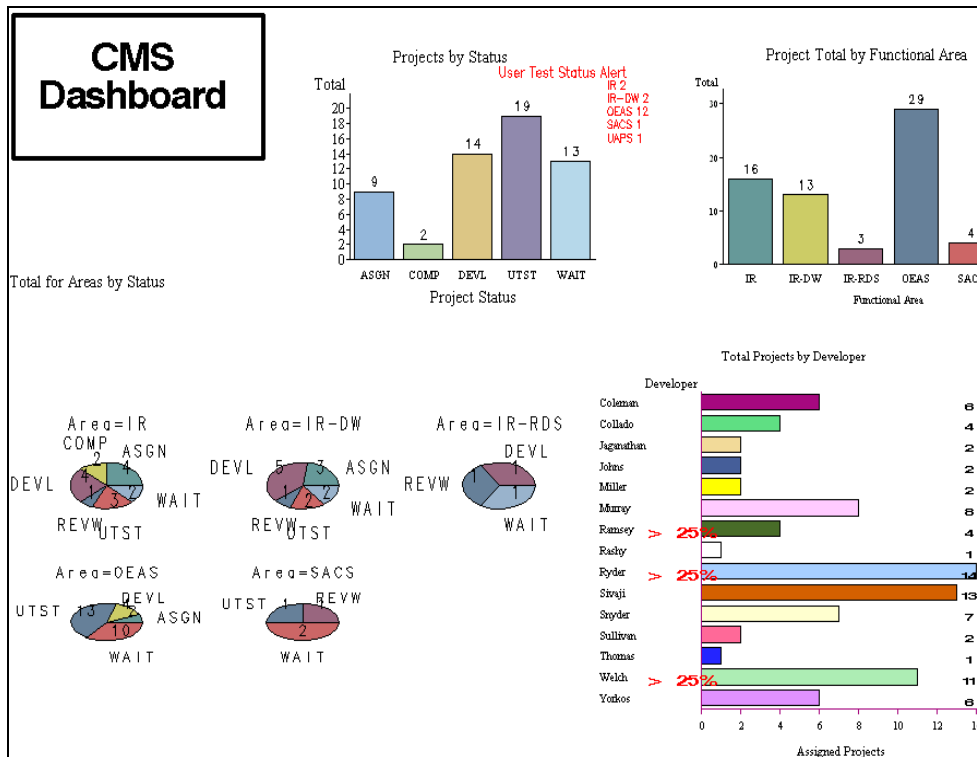


Figure 3: Programmer 2 dashboard display



On-going Development

Neither of the designs in Figure 2 or 3 are production-ready. Following a functional review, several design alterations were requested and pieces of both Programmer 1 and Programmer 2 designs were incorporated into a second generation output.

MAINTENANCE AND FUTURE ENHANCEMENTS

The initial design of the CMS IR Technical Support Dashboard and its easily manageable data structure enables efficient maintenance by the developers. KPIs, alerts or all other data structure changes on any object on the dashboard can easily be implemented without affecting other objects in the same environment. One enhancement that has already been identified is the incorporation of "Due Date" alerts (i.e. Jobs with today's due date, Jobs that are past due).

Subsequent to our initial designs, the SAS BI Dashboard 3.1 functionality was released. We are currently evaluating the new product and plan to convert to pre-existing technology to the new drag-and-drop tool.

CONCLUSION

The CMS IR Technical Support Dashboard joined together functional project managers and technical developers to provide a solution for more efficient resource allocations. The dueling output options provided by two independent developers allowed for maximum presentation flexibility and a training opportunity for both programmers. The current designs of the dashboard provide graphical displays with quick-look views to identify successes, challenges and overburdened technical staff, teams or offices. The final product increased the ease of overall project management and specific-project-level oversight.

REFERENCES

SAS/GRAPH Dashboard Samples at <http://support.sas.com/rnd/datavisualization/dashboards/index.html>

CONTACT INFORMATION

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

M. Paige Borden
Director and University Data Administrator
Office of Institutional Research
University of Central Florida
P.O. Box 160021
Orlando, FL 32816-0021
Work Phone: (407) 823-5061
Fax: (407) 823-4769
Email: mborden@mail.ucf.edu
Web: <http://www.iroffice.ucf.edu>

Maureen Murray
Coordinator, Computer Applications – Data Warehouse Team
Office of Institutional Research
University of Central Florida
P.O. Box 160021
Orlando, FL 32816-0021
Work Phone: (407) 823-4764
Fax: (407) 823-4769
Email: mamurray@mail.ucf.edu
Web: <http://www.iroffice.ucf.edu>

Ali Yorkos
Coordinator, Computer Applications – Data Warehouse Team

Office of Institutional Research
University of Central Florida
P.O. Box 160021
Orlando, FL 32816-0021
Work Phone: (407) 882-
Fax: (407) 823-4769
Email: ayorkos@mail.ucf.edu
Web: <http://www.iroffice.ucf.edu>

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.