Paper 2621-2018

Improving Scheduling and Strategic Planning at JPMorgan Chase's Card Production Center

Jay P. Carini and Joel C. Weaver, JPMorgan Chase & Co

ABSTRACT

JP Morgan Chase's manufacturing-based card production center (CPC) is beginning to leverage their vast array of production data and take advantage of the benefits provided through analytics using end-to-end SAS®-based solutions.

Due to the complex nature of their operation, CPC desired a more efficient process to support strategic planning and scheduling, strategy related to unit cost and revenue generation, and optimizing resource allocation. Utilizing the CPM Procedure within the SAS/OR® suite, we built a model which allows the business to schedule tasks and test various strategic scenarios subject to priority, resource and time constraints, across a variety of products within their desired planning horizon.

In order to operationalize the solution and demonstrate business value before investing in costly IT resources, we developed a user interface integrating SAS stored processes, the SAS Add-In for Microsoft Office (SAS AMO) and customized VBA code. The user interface allows our non-technical users to harness the power of SAS modeling, submit customized scenarios, run the model, and receive customized output – in an on-demand, user friendly environment.

The project demonstrates the importance of starting small and focused when introducing analytics into a new environment. The success and business buy-in from this initial project is leading to additional refinements and an expansion of the model capabilities to address new use cases and new business units.

INTRODUCTION

The Card Production Center is a manufacturing environment requiring careful consideration of various factors to produce, personalize, and mail credit and debit cards to customers, including:

- Card material
- Service level agreements (priority)
- Job size
- Type of card (credit vs. debit)
- Potential production locations

Figure 1 provides an overview of the production process. Cards are requested as part of "Jobs". Each job passes through three major production steps including:

1. Embossing – personalizes card with customer and account information, and encodes the chip or magnetic strip. These machines are specific to the card material (plastic, embedded metal, and metal)
2. Match Merge – attaches the card to the paper carrier (the piece of paper with customer information and potentially product/marketing material)
3. Insertion – Inserts the carrier (with card) into the envelope, and seals envelope

Depending on the product, CPC also utilizes a custom machine which combines the Match Merge and Insertion steps. Each of the steps above requires at least one operator throughout its process.

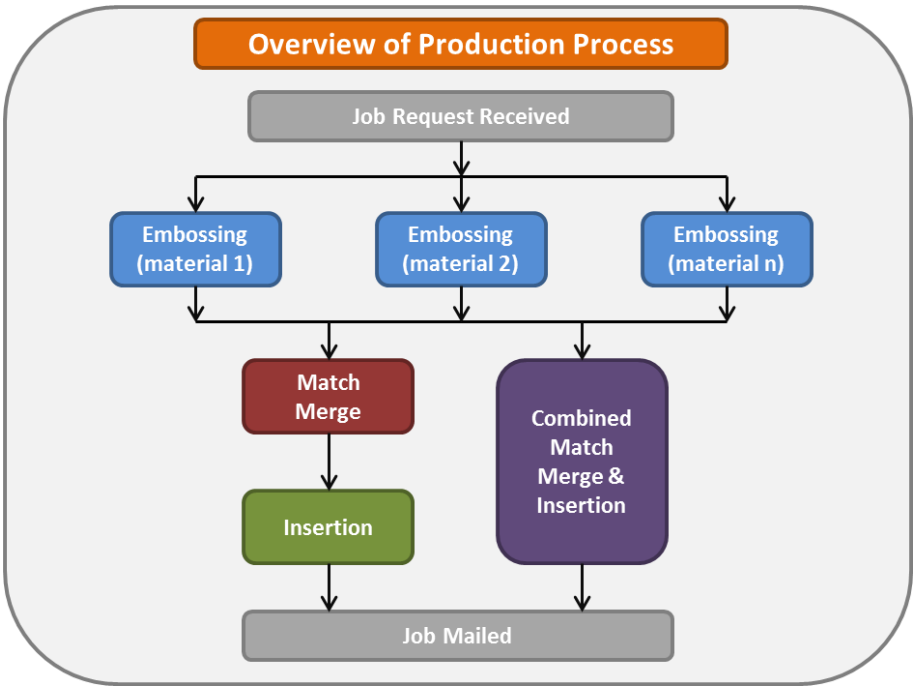

Figure 1 Overview of CPC Production Process

Prior to this project, the CPC used ad-hoc scheduling tools (i.e. spreadsheets), management expertise, and intuition to plan and accept requests for future jobs. CPC engaged the Chase Operations Research and Data Science Center of Excellence (ORDS CoE) to develop a systematic, user friendly, data driven capacity management and scheduling tool to support strategic planning and scenario testing.

METHODOLOGY

After researching a number of potential solutions including traditional simulation packages often used in manufacturing environments and external vendors/standalone scheduling tools, the ORDS CoE utilized the CPM procedure within the SAS/OR suite as the basis for the project.

The key features of Proc CPM align closely with the objectives of the project: scheduling tasks subject to priority, resource, and time/calendar constraints.

Five input datasets were developed in the execution of Proc CPM:

- Tasks – the main dataset which defines the input tasks including their priority, due date, required resources, required time, and required steps
- Resources – specifies the available resources, including machines and operators, the dates/times they are available, and relative machine performance to be used with the *Resource-Driven Duration* specification.
- Shifts – specifies the unique shifts which can be assigned in the calendar dataset
- Calendars – assigns shifts by day of week to create unique calendars
- Holidays – specifies holidays where the operation will be closed, regardless of other calendar/shift assignments

We used the following code as the basis for invoking the CPM procedure:

```
PROC CPM DATA=SASTasks holidata=SASHoliday out=DetailedOut
interval=dthour workdata=SASShifts
resin=SASResources resout=ResourceOut resourcesched=ResourceSchedule
calendar=SASCalendar;
id NumCards;
activity task;
succ succ1 / lag=Lag;
calid cal;
project project;
duration NumHour;
aligndate adate;
aligntype atype;
res Embossing MatchMerge Insertion Combo Vault PriorityMail
Operator /  period=per obstype=otype delayanalysis rescalintersect
resid=resid work=work schedrule=actprty
schedrule2=lst ACTIVITYPRTY=priority addcal;
holiday  holista / holiend=(holiend);
run;
```

TASKS DATASET

Requests for new cards come to CPC in the form of "Jobs"; each of these Jobs includes a number of cards (ranging from 1 card to 100K+ cards depending on the job type). We modeled the operation at the Job level, with each Job corresponding to up to seven records in the Task File:

1. Vault – the process for counting and pulling the correct number of "blank" cards from the vault
2. Embossing – as described in the Introduction
3. Quality – a quality checkpoint before the cards move on through the process
4. Match Merge – as described in the Introduction
5. Insertion – as described in the Introduction
6. Mailing – certain jobs require an extra step to insert the envelope into an expedited shipping package (i.e. FedEx or UPS).
7. Job Summary – record used to name the Job, used in coordination with the "Project" option in the Proc CPM code to allow for easily summarizing each job in the output datasets

We used a number of fields in the Task file, to both accurately represent the characteristics of each Job, but also to help identify the Jobs in the output data sets. A few of the key variables (`with the variable name corresponding to the code above`) in the task dataset include:

- Calendar (`Cal`) – providing option for assigning different types of tasks to different calendars/shift types, for example the vault opens before the rest of the shop, allowing Vault tasks to be completed at an earlier time.
- Task (`Task`) – the unique name of each task
- Time required for task completion (`NumHour / Work`) – based on the job type and task step, each task was assigned a completion time based on expected throughput. The `Work` variable was used to allow for Resource-Driven Durations.
- Successor (`succ1`) – Each task was assigned a unique successor
- Start and End times for each job and their alignment types (`adate` / `atype`):

    - The first task of each job (Vault) was assigned the earliest date/time the job could start, with an align type value of SGE (start greater than or equal to)
    - The last task of each job (Insertion or Mailing depending on the job type) was assigned the deadline for the job, with an align type value of FLE (finish less than or equal to)
- Resource Columns – a column corresponding with each machine and operator resource.
    - For machine columns – a 1 was placed in the column if the corresponding task record requires that machine
    - Some machines were represented as Machine Groups – including multiple machines, as defined in the resource dataset
    - For operator columns – the number of operators required for the task was entered
- Job Priority (`Priority`) – specifies the relative priority of each job, to ensure accurate scheduling under the scheduling rule: `ACTPRTY`.
- Project (`Project)` – Each task of the Job was assigned a Job name, to allow for more efficient identification in the output files.
- Various job demographics, including number of cards, job type, and client name, etc.

RESOURCES DATASET

The resource dataset specifies the important characteristics of the machines and operators in the operation. Four different `otype` variables were used in the dataset, including:

- RESRCDUR – specifies the effect of resource on project time. We used "1" for all machines, i.e. machines are Driving Resources. We used "2" for all operators, i.e. operators are spanning resources required for the duration of the task.
- ALTRATE – Used in the creation of Machine Groups. The values in these records identify the relative performance of each machine in the group. The values in these records are applied against the WORK duration in the task dataset to determine the actual duration of the task given the selected resource.
- ALTPRTY – Used to prioritize each machine within the machine groups. These priorities were determined by consulting with the operation to determine their preferences within each group.
- RESLEVEL – Specifies the number of resources available at each time specified in the PER column. We included a RESLEVEL record for every 1-2 hours, reflecting the actual fluctuations of operator levels throughout the day.

SHIFTS, CALENDAR, AND HOLIDAY DATASETS

These three datasets were used to set the calendar constraints within the operation.

- Shifts – specified three different categories of shifts for Operators, the Vault, and the machines. The shifts were used to reflect the operating hours of the business and expected break times for the operators. The Vault was broken out separately since it typically opens earlier than the rest of the operation. Using Operator shifts, we were able to build in expected breaks throughout the day.
- Calendars – built using the shifts, again a separate calendar for Operators, the Vault, and the remaining machines.
- Holidays – reflect the days the entire operation is closed.

OUPUT DATASETS

Three main output datasets were provided by Proc CPM and used in this project, including:

1. OUT
2. RESOUT
3. RESOURCESCHED

These datasets were manipulated to provide key metrics back to the business. For each scenario the business requests through the tool they are interested in three key metrics:

1. Number of Jobs that missed their Service Level Agreement (SLA)
2. The amount of expected "Slack" for jobs completed on time – the amount of time between the job completion and the deadline, used to understand the amount of risk that may exist in a schedule
3. Machine and Operator Utilization – the percent of available time a machine or operator is used on tasks. This metric is used to help management plan for future resource levels to ensure machines and operators are being used efficiently.

MODEL VALIDATION

ORDS CoE validated the model across a number of dimensions, including cards produced by day, machine utilization, operator utilization, and SLA adherence. Figure 2 below summarize results of the model back-tested against actuals, comparing the number of cards produced across the three major production steps (Embossing, Match Merge, and Insertion).

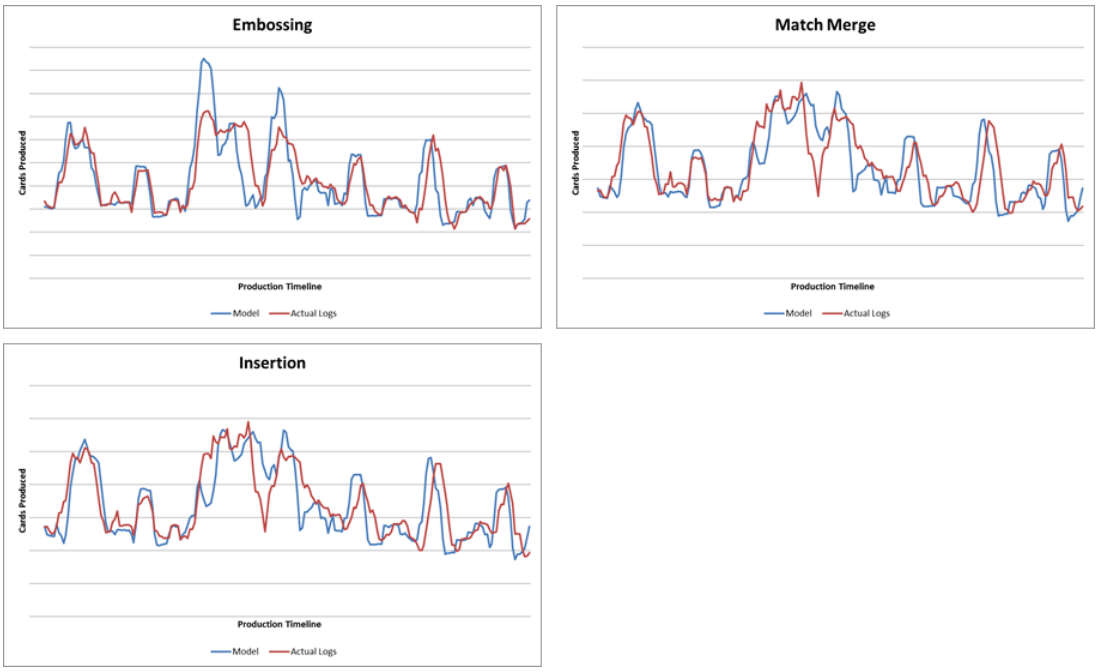

Figure 2 Summary of Validation/Back Testing

The validation performed well over the test period and provided the business a high level of confidence that the model could be used to support strategic planning. The deviations between the model and actual performance are expected and are due to the model not fully accounting for daily operational "fluidity" (machine downtime, operator absence, purposeful level-loading, etc.)

USER INTERFACE

Providing end users access to the scheduling tool required a simplified interface that did not rely on the users having prerequisite knowledge of SAS programming or other SAS tools (i.e. SAS Enterprise Guide®). Microsoft Excel was chosen as the platform for building the model UI for multiple reasons:

- Microsoft Excel is a familiar environment for most business users.
- Data entry can be controlled using Excel UserForms.
- In conjunction with VBA coding, the UserForms can reduce user input errors and provide a structured path for the user to take (an input Wizard).
- Microsoft Excel utilizes the SAS Add-in for Microsoft Office (SAS AMO), which provides users access to data on the SAS Server from their PC.

Figure 3 outlines the scenario request process; from submitting the scenario request details to receiving the corresponding email summary output.

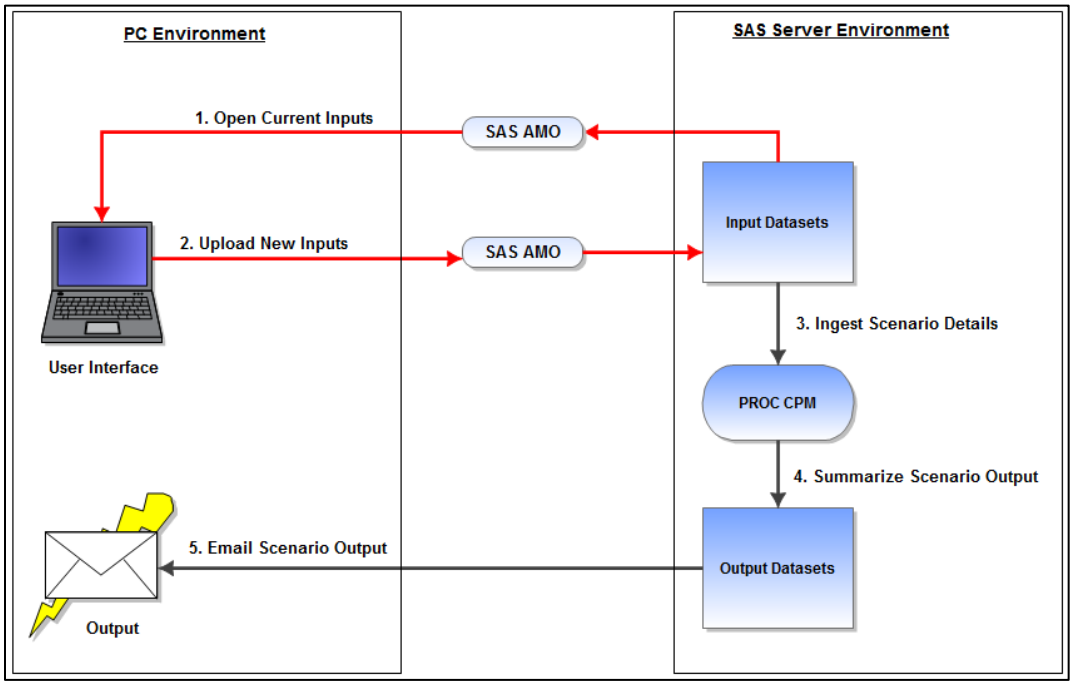

Figure 3 Scenario Request Process Flow

Users begin by selecting the type of request they would like to submit along with other basic request details (Figure 4). Next, they specify the input parameter(s) they would like to change for the specific scenario they are running (Figure 5).

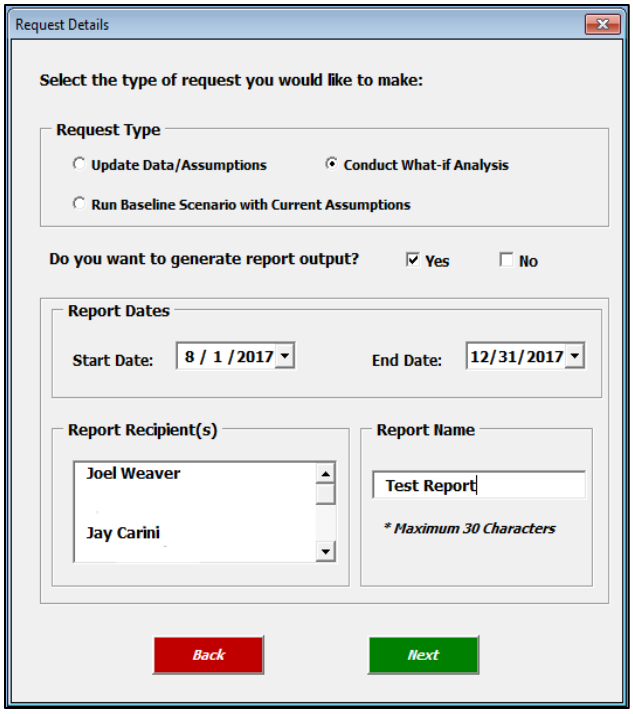

Figure 4 Example UserForm for Initial Request Details

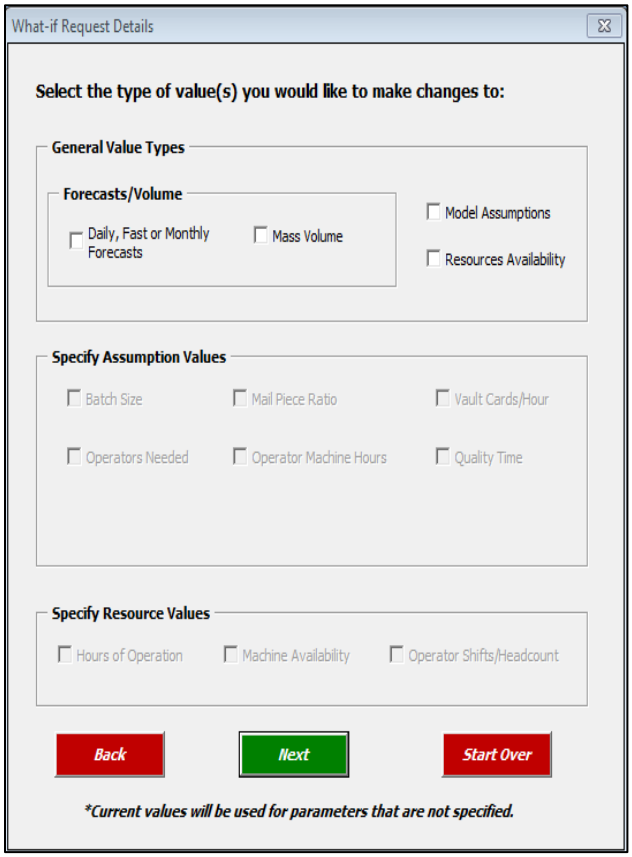

Figure 5 Example UserForm for Request Parameters

In step 1 of the process flow (Figure 3), the user “opens” all applicable input datasets based on the scenario parameters they specify. The datasets that are being changed are not opened in a traditional sense. The applicable dataset is inserted into a blank Excel worksheet using VBA code tied to active buttons on the Excel UserForm. Specifically, the VBA code utilizes the SAS AMO to act as a bridge between the SAS server environment and the PC-based user interface. The sample VBA code below illustrates the process for accessing the input data from the SAS server (step 1):

```
Dim sas As SASExcelAddIn

Set sas = Application.COMAddIns.Item("SAS.ExcelAddIn").Object

Dim data As SASDataView

Set data = sas.InsertDataFromLibrary(Server As String , Library As String, Dataset As String,
OutputLocation, PageSize As Long , DisplayAllRecords as Boolean, Filter As String , Sort As
String , ShowRowNumbers As Boolean)
```

Once the dataset is inserted into a new worksheet, the data can be viewed or referenced in any relevant UserForm. Figure 6 is an example UserForm that can be used to update the shift/headcount parameters for a specific scenario request:

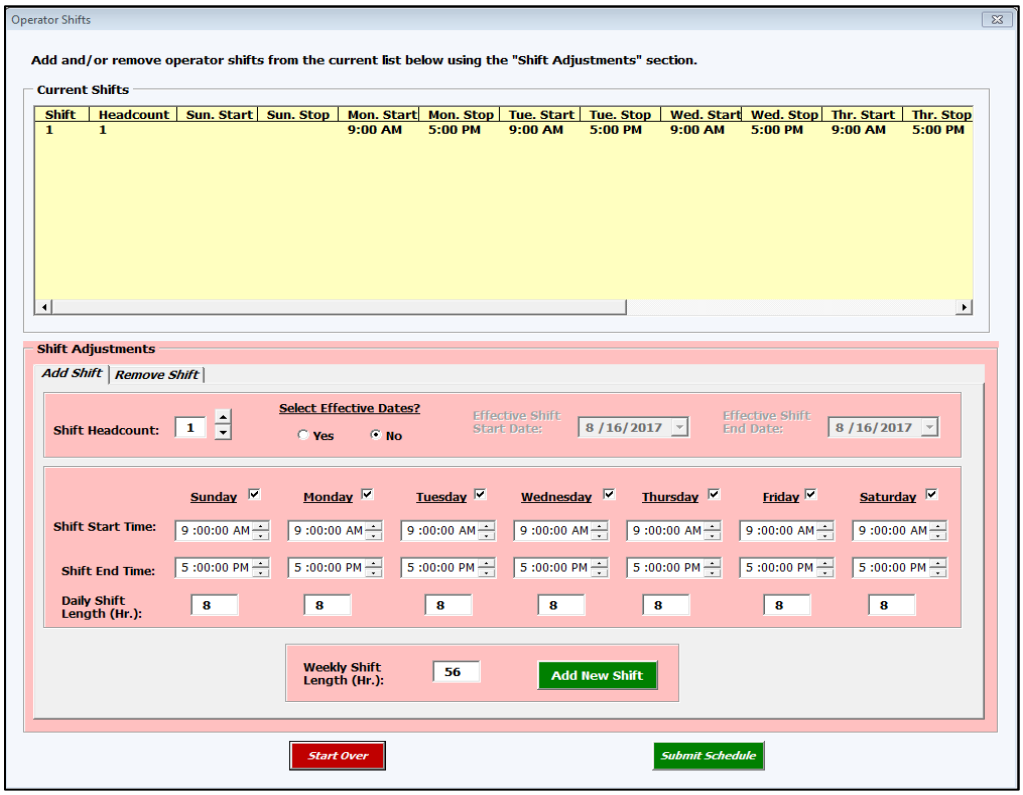

Figure 6 Example UserForm for Specifying Shift and Headcount

Upon completing the desired changes, the user clicks submit on the UserForm to complete the upload process to the SAS server (outlined in step 2 of the process diagram, Figure 3). The upload process is completed using the SAS AMO to call a SAS Stored Process that formats and transforms the data prior to model ingestion. The sample VBA code below shows how to invoke a SAS Stored Process:

```
Dim sas As SASExcelAddIn
Set sas = Application.COMAddIns.Item("SAS.ExcelAddIn").Object
Dim stp As SASStoredProcess

Dim prompt As SASPrompts
Set prompt = sas.CreateSASPromptsObject
prompt.Add "prompt1", <prompt1_val>
prompt.Add "prompt2", <prompt2_val>

Dim inputdata As SASRanges
Set inputdata = New SASRanges
inputdata.Add "inptdata", <Worksheet Range>

Set stp = sas.InsertStoredProcess(path As String, OutputLocation, [ prompts],
[outputParameters], [inputStreams])
```

The input data name identified above ("inptdata") needs to be the same name used in the libname statement in the stored process being called. The sample SAS code (initial part of Stored Process code) below shows how to reference the data in the specified Excel worksheet range:

```
libname inptdata xml;

DATA fromExcel;
set inptdata.&_WEBIN_SASNAME;
run;
```

After completing the edits and uploads for all selected parameter inputs (Figure 5), the process can continue with steps 3 – 5. As part of steps 1 and 2 in the UI process, a summary dataset keeps track of all request details to provide the appropriate inputs to the SAS script containing PROC CPM. Based on the parameter inputs, the scheduling model runs and generates the appropriate output. The model output is generated using several SAS procedures (PROC SGPLOT, PROC REPORT, PROC TABULATE) and emailed to the specified report recipient(s) (Figure 4) via the SAS output delivery system.

CONCLUSION

This project has several observed benefits and has the potential to offer more with future enhancements. Some of the observed benefits are:

- Enables quick assessment and scenario generation to find best-fit options to meet client needs - old tools required hours or days to respond to clients, the new model provides more accurate guidance within minutes
- Quickly recasts entire job-plan when additions, deletions, and changes are introduced
- Evolves management behavior towards reliance on data and automation to govern daily operations

The modeling output can be further enhanced by leveraging dashboards constructed in SAS Visual Analytics as a replacement for the current canned emailed output. Potential benefits of improving the output would be:

- Identifying production constraints or potential break-points, allowing the business to institute risk controls, expand capabilities to ease the constraints, or stretch the failure threshold
- Provide better views for comparing performance across different scenarios

USE CASES

Initially the business has applied the model towards four general use cases:

- New high priority requests - model estimates earliest completion dates, and the impact on completion dates for previously committed jobs
- New low priority requests - model estimates the earliest completion dates without impacting prior commitments
- Resource scenarios - impact of adding/dropping operators/machines, utilizing overtime, machine downtime
- Location scenarios - impact of shifting work between locations, disaster recovery testing

Figure 7 demonstrates an example use case of the model. Three Gantt charts represent three different scenarios:

1. Baseline – the current book of work
2. Scenario 1 – Business requests a new job with medium priority
3. Scenario 2 – Business requests a new job with the highest priority

Each row of the Gantt charts represents a machine, and the colors in the rows represent the completion of jobs with differing priorities. Red are High, Blue are Medium, Orange are Low, and Yellow represent a new request.

In Scenario 1, the New Job is completed after the higher priority existing jobs. In this example the new request is completed in the middle of Day 3. In Scenario 2, though, since the new request has the highest priority, the completion times of other jobs are shifted, while the new request is completed in the morning of Day 2.

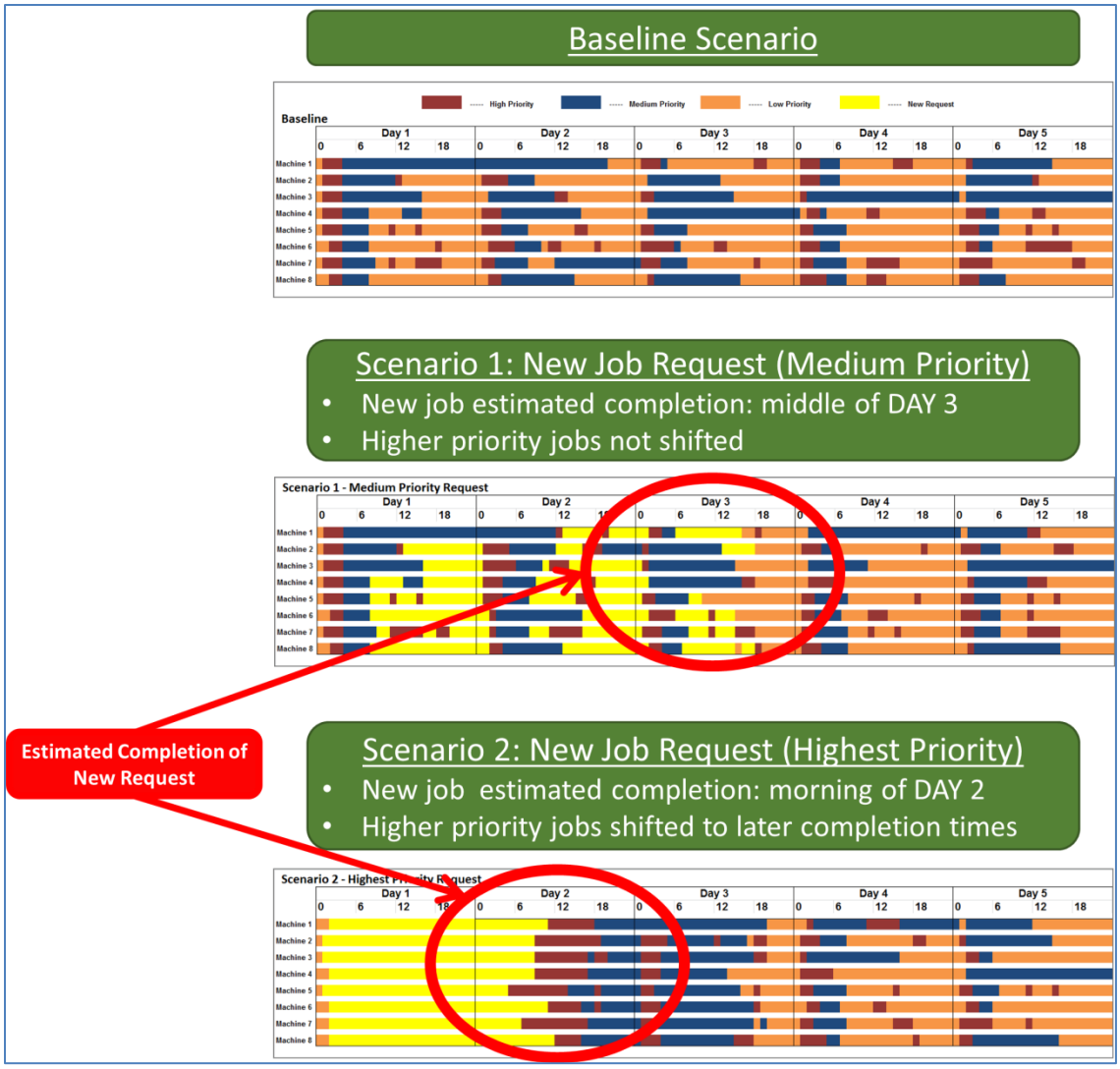

Figure 7 Use Case Example

REFERENCES

Beese, Tim. 2011. "Tips and Tricks for Automating the SAS Add-In for Microsoft Office using Visual Basic for Applications". Proceedings of the SAS Global Forum 2011 Conference, Cary, NC: SAS Institute Inc. Available at http://support.sas.com/resources/papers/proceedings11/012-2011.pdf

Beese, Tim, Granger, Greg. 2012. "Excelling with Excel". Proceedings of the SAS Global Forum 2012 Conference, Cary, NC: SAS Institute Inc. Available at http://support.sas.com/resources/papers/proceedings12/036-2012.pdf.

SAS Institute Inc. 2011. "Sample 42983: Using SAS Add-in for Microsoft Office scripting to process Microsoft Excel data and display results". http://support.sas.com/kb/42/983.html

CONTACT INFORMATION

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

Jay P. Carini
JPMorgan Chase
jay.p.carini@chase.com

Joel C. Weaver
JPMorgan Chase
joel.c.weaver@chase.com