

# A User-friendly Excel Database Management Toolkit to Model and Optimize an Arbitrary System

Soha Al-Refai†

†Faculty of Computer Science, The Public Authority For Applied Education & Training, Hawally, Kuwait

## Summary

Microsoft Excel in conjunction with Simio's Risk-based Planning and Scheduling (RPS) algorithm can be used to create a user friendly scheduling tool. By coding a data-collecting toolkit using Visual Basic (VBA), any company or organization can easily maintain records of all entities going through their system. This software then uses the collected information as inputs for a Simio model of the system and returns the approximate order waiting time.

### Key words:

Database, Management, Simulation, System, Model, Schedule

## 1. Introduction

With over 55% of all small scale businesses (SSBs) and organizations having no website, there is a need for an efficient and simple way to digitize services [1]. By digitally backlogging past transactions and cataloging the movement of products within inventory, SSBs can improve their service times dramatically. In extension, this would lead to higher customer satisfaction.

The creation of a user-friendly tool that would easily convert physical systems into digital ones would also allow for rapid improvement in manufacturing facilities. Bottlenecks in production can be easily identified and addressed if there existed a method to accurately keep track of material flow [2]. For this toolset to be even more beneficial, it should work alongside Simio's RPS technology. This uses all digital data that gets backlogged to predict a schedule that maps both the movement of parts within a facility as well as the estimated completion times.

## 2. Background

A system, like that of a fast food restaurant, can be described as a set of processes that are necessary to meet a pre-defined goal. Using this definition, a model of said restaurant could include the steps of taking, preparing, and serving a meal in order to meet the goal of customer satisfaction by minimizing the waiting time. As visualized

in Figure 1 below, this is an example of an M/M/c queue where  $c = 1$  server.



Fig. 1 M/M/1 Queue

Moreover, systems can be more complex with additional servers and a certain degree of variability. The fast food server, for example, may be new and would thus need more time than average to take an order. Another example could be a faulty appliance in the kitchen that is known to break down on occasion; thus there would be an impact on the food preparation time. In acknowledging these sources of variability, a more accurate model would need to be used.

A G/G/c queue takes this into account by representing a system where interarrival and service times share a general distribution. In order to accurately model and predict a G/G/c queue, however, detailed records of time data should be available. The Kingsman Diffusion Approximation is an algorithm that uses this data to predict the mean waiting time of any entity within a G/G/1 queue [3]. A simulation software called Simio uses this algorithm along with others to generate a schedule of the system processes and point out risk areas within the queues.

## 3. Approach

### 3.1 Establishing an Arbitrary System

In order to efficiently code an excel based scheduling software and a Simio simulation model, a predefined system has to exist. As such, the flow figure below describes the system that will be used for this paper.

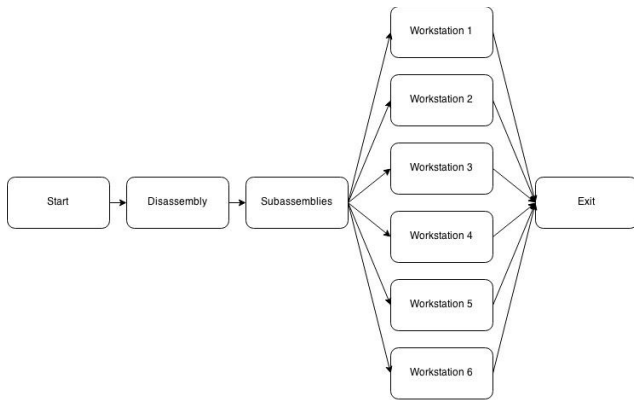


Fig. 2 An Arbitrary System

Products entering will first be fully disassembled. Then they will be moved and reworked in a subassembly area. Finally, the products will be sent to one of six available workstations to be completely reassembled. Though this system will be the example used for the software described in this paper, the same methods can be reapplied to work with any other system.

### 3.2 Risk-based Planning and Scheduling

There are two well-known scheduling approaches that could be modeled. The traditional Advanced Planning and Scheduling (APS) approach requires that all the data needs to be fully known and deterministic. For example, all processing times must be fixed without variations and there can be no unexpected events or delays. The resulting schedule with APS should therefore represent a perfect facility which is unrealistic because there will always be discrepancies in a real facility. Predicted schedules and the actual performances will also be significantly different if the scheduling engine removes all variations from the schedules.

Traditional APS also forces people to work with a perfect schedule which means they will always promise more than they can deliver when it comes to customer requirements. An experienced scheduler should know that a perfect schedule is seldom met. As a result, the scheduler is forced to add buffers such as extra time, extra inventory, or extra capacity. This causes variation which results in inefficiency and higher costs. APS does not provide the critical information needed for determining what combination of these buffers is necessary to produce a robust schedule while minimizing the cost.

Simio uses the Kingsman Diffusion to fully account for variations in production systems as well as providing the scheduler with the right information to reduce risks and uncertainty. The simulation model used in the RPS approach can be built in great detail and can incorporate any random variations that are present in a real system. Although the usage of RPS is relatively new, there is no

doubt that it has been successfully used in a wide range of application areas. These simulation models are frequently used for evaluating production lines and facility designs. Simulation allows anyone to make changes in the model while testing out new methods before implementing these methods into the real system.

A Simio simulation model allows people to easily analyze and compare system designs to reduce cost and improve the system performance. Furthermore, the daily operations of a facility can always be improved with the new RPS application. This is extremely beneficial for businesses because it allows the businesses to set realistic schedules that account for uncertainty. It also allows businesses to determine risks early and formulate strategies that can produce revenue while reducing costs with existing assets [4].

## 4. Methodology

### 4.1 Excel Dashboard

An Excel dashboard had to be created as a tool in order to control the inputs to the simulation model and monitor the product assembly process. It is extremely important for this dashboard to be as user friendly as possible and for it to have the ability to display the results of the simulation in real-time. This allows organization managers to make the appropriate decisions regarding the employee schedule needed for products to be completed on time. The dashboard was designed using VBA code and includes several user forms that are used to control all the inputs to the simulation. Although many different worksheets are available, everything is simply and easily controlled from the “Dashboard” worksheet.

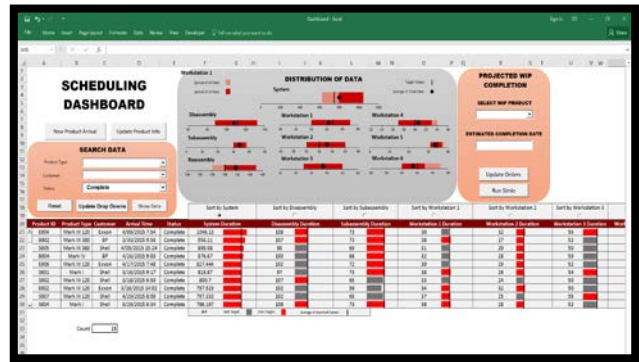


Fig. 1 Excel Scheduling Dashboard

Upon opening the Excel file, the first worksheet to be displayed is the “Dashboard”. Here, drop down menus allow for the display of products by product type, product status, and customer in a table underneath. Small charts for each product are given which show the current status and

can be arranged by workstation. If the chart is colored in grey, the Product is meeting the target time, if it is in red it has gone over the target time. A line in each chart also shows the average time of the searched values. Above, another display shows the distribution of data for all tables. This is given for the entire product assembly process and for each individual workstation. These tables also show the spread of the 10 data points given by the searched value, the spread of all the data, the target value, and the average total data.

Located to the right of the Distribution of Data chart, there is a box labeled Projected Work in Progress (WIP) Completion. After running simulations with the current products in the system, Simio determines estimated completion dates. These can be displayed for each individual product in the Projected WIP Completion Box. This information is very useful to the employees at an organization because it allows them to allocate resources efficiently in case the priority of the products they choose is higher than normal.

To input a new product coming into the system, a button titled “New Product Arrival” was placed above the search data. This opens up a user form which allows you to enter new Products and submit information such as Product ID, Customer, Product Type, Arrival Date, Arrival Time, Target Completion Time, and Target Completion Date. Upon submitting the new Product, it enters the database and can be called up when necessary.

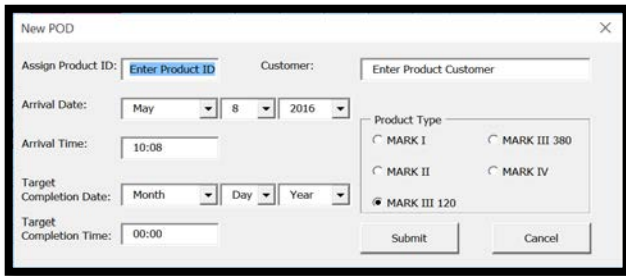


Fig. 2 New Product Userform

As products move between workstations, the user must update the information by pressing the “Update Product Info” button also located above the search data window. Upon doing so, a time stamp of the current time and date will be saved and thus inter arrival time between workstations can later be automatically calculated. Here, another user form is displayed which allows users to record new data for the product. It updates the status of the product and identifies in which step of the assembly process it is currently at. Upon pressing “OK”, the particular product information is updated in the datasheet.

As the information is entered into the dashboard, it automatically updates on other worksheets in the file. Other worksheets such as WIPProduct, OrderData, OrdersTables and ProcessingTimes are all used to control the simulation model. After 25 Products of each type have passed through the system, the average inter arrival times are calculated and entered into the database to be used by Simio.

#### 4.2 Simio Model

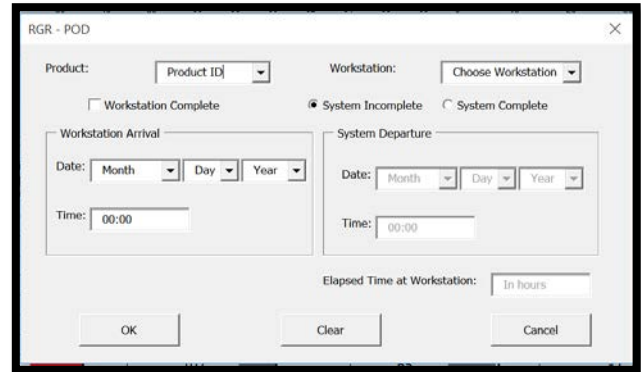


Fig. 3 Update Product Userform

In order to depict the assembly layout described by the Excel software, a source, eight servers, and a sink were placed into a Simio model. A model entity was also placed to act as incoming product orders. With the source representing the total demand for products, an exponential distribution is assigned based on the inter arrival time data taken from the Excel dashboard.

Connected to the source’s output is the first server, being the product Disassembly. Following this is another server that acts as the product Subassemblies. From here, the model breaks off into six different paths feeding the six remaining servers. Finally, these six workstations all lead to the sink which represents the product shipment site. The processing times for all servers in this model were entered following an exponential distribution. This way, even with the limited time data, the model would still take variability into account.

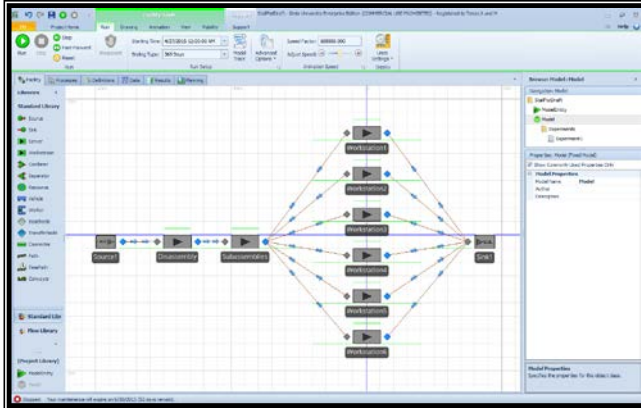


Fig. 4 Simio Model

At the product Subassemblies output node, there are six connectors that each attach to one of the six workstations. When running the simulation, the default settings of Simio would randomly select between the attached servers as the respective product’s path. By allowing for this happen, the simulation would inaccurately move multiple products to the same workstation even if other servers may be available. To remedy this, a list of server nodes had to first be created. This list points to the locations of the six workstation inputs.

Node
Input@Workstation1
Input@Workstation3
Input@Workstation2
Input@Workstation4
Input@Workstation5
Input@Workstation6
*

Fig. 5 Server Node List

Having defined this list, Simio can then able to adjust the routing logic at the subassemblies output. Entity Destination Type was set to “Select From List” and thus identifies the recently defined list of server nodes as a reference. This unlocks a setting entitled Selection Goal where “Smallest Value First” was chosen from a drop down list of options. In order to have the routing selection be based on the input with the least amount of waiting products, a Selection Expression had to be clarified. The expression “Candidate.Node.AssociatedStationLoad” was entered referencing the load associated with each station. At this point, the model is almost complete. A constraint of one has to be defined as the capacity of each operator. The reason is, in defining constraints of one operator per workstation, the model will be flexible enough to make use of Simio’s OptQuest feature. This is a critical tool

because it will provide a basis for which optimized product assembly times can be outputted and taken to be used in the excel dashboard.

### 5. Conclusion and Recommendations

The Excel software that has been described is limited only to the specific system established for this paper. Although this is a major drawback, the toolset works excellently in conjunction with Simio’s RPS algorithms to map out material flow and identify bottlenecks.

In order for a facility to realistically adopt this, however, some of the code would have to be changed. Depending on the amount of workstations and products involved, the Simio model itself would also have to be changed. A final limitation is the fact that Simio and Excel both charge enterprise fees which may end up being too costly.

Nonetheless, this paper can serve as an outline for creating custom excel dashboards that could work for any system. Improvements could include a start page on the dashboard that would allow a user to define his/her own system. This way, even without having any technical knowledge, the tool would be readily available to any business or organization that is willing to use it.

### References

- [1] D. Ickert, "2013 Small Business Technology Survey," National Small Business Association, Washington DC, 2013.
- [2] S. Minner, "Forecasting for Inventory Management of Service Parts," in Service Parts Management, Springer, 2007, pp. 157-169.
- [3] R. M. F. Guy L. Curry, Manufacturing Systems Modeling and Analysis, Second Edition ed., New York: Springer, 2011.
- [4] D. Pegden, "Business Benefits of Simio's Risk-based Planning and Scheduling (RPS)," Simio LLC, Pittsburgh, 2011