A Special Report:

How Leading Discrete Manufacturers Minimize Order Delivery Schedule Risk & Associated Costs In Advance - Automatically

Automatic Comparison to Current Plan
Instant Probability Prediction for On-Time Delivery
Guidance for Re-allocation of Resources - Re-Plan Quickly - On-the-Fly

By C. Dennis Pegden, Ph.D.

Executive Summary

What if your current dashboard reporting system could automatically compare your current schedule to a future time, and accurately forecast the probability of individual order deliveries – enabling corrective action before issues become problems?

Until recently, unavoidable variations in production made this impossible. No matter how strong the original plan, these variations made every schedule infeasible over time, resulting in production delays or unanticipated costs.

However, now there is a new methodology, simulation-based operational analytics (SOA), that can account for the underlying risk imposed by variations in the system. These new analytic tools go beyond the traditional use of simulation for assessing alternative designs. Instead, they directly support the use of models within an operational setting to improve the everyday production, operational, and business level decisions that are key drivers to the overall success of a discrete product manufacturing plant.

Simulation-based Operational Analytics (SOA) is the application of simulation methodology to operational planning and scheduling. The basic concept is to leverage the predictive power of simulation models to improve the daily operations of a system.

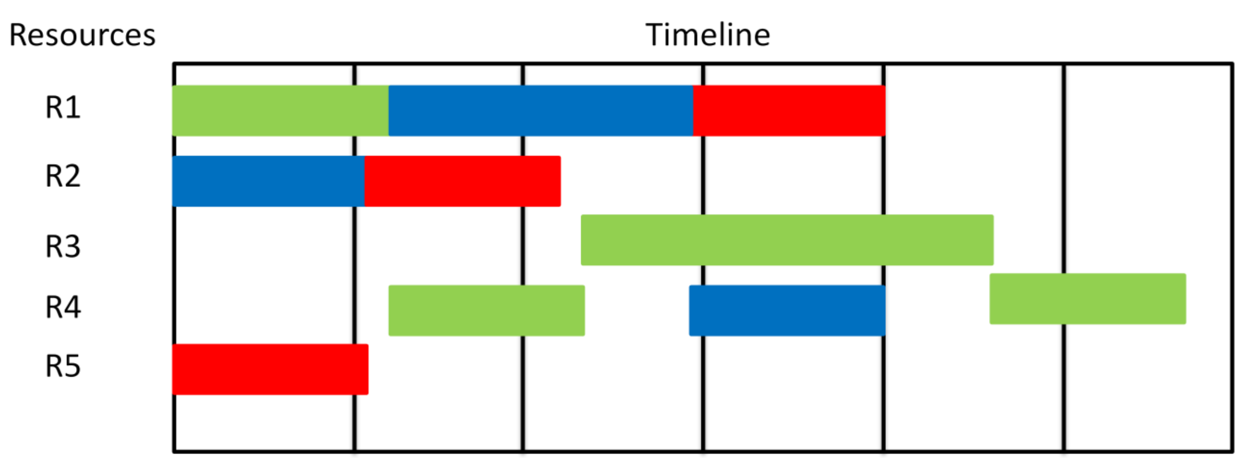
In a traditional simulation application, the model compares alternative system designs to make improvements to the system. For example in an assembly plant application, you might use a simulation model to determine the number and type of each machine at your workstations, as well as learned operator skills, material delivery, and production strategies. Once you have designed your production system, you are done with the model until you revisit the design at some point in the future. In these traditional simulation applications, you only make use of the model on an occasional basis, when evaluating fundamental changes to the underlying system design.

SOA provides value in day-to-day planning and scheduling

In contrast, with SOA, you can use your model on a daily basis to help re-schedule your system operations on-the-fly. Hence the model delivers value on a continuous, ongoing basis. The basic purpose of the SOA model is to determine the best sequence for a set of tasks across a limited set of resources. In an assembly manufacturing operation, you could use an SOA model to compare the master plan and schedule to the actual production of orders, based on current conditions in your facility. Your model could be used on an ongoing basis to forecast risk, cost and quality of the individual order delivery schedule days -- weeks or months prior to the actual work. This gives you more time to mitigate the risk while managing the cost of change.

In an operational setting, the SOA model works with actual data for individual transactions. In an assembly plant application, the model would process a specific list of production orders, using actual routings and expected processing times, expected material arrival dates, etc. This data is typically downloaded to the model from the MRP or ERP system. Although there is typically variation in things like processing times, material arrival dates, etc., the planning and scheduling is done with all deterministic values. All randomness that is normally present is removed, and all times are assumed to be their expected values. Likewise, all unplanned events such as machine breakdowns, workers calling in sick, etc., are eliminated from the SOA model execution. This is necessary because it is not possible to develop a detailed plan or schedule that incorporates variation and unplanned events.

The output from the traditional SOA model is often viewed in the form of a Gantt chart that shows individual transactions across resources and over time. In the following simple example, there are three orders to assemble (green, blue, and red) that are processed across five workstations (R1, R2, R3, R4, and R5). The red order is first processed on R5, then R2, followed by R1. The simulation model generates this production plan by simulating the actual movement of these three orders through a detailed model of the limited resources in the system. The model logs the start and stop time for each order on each resource, and these times are then used to display the schedule in the Gantt chart.



To generate this schedule the model assumes all deterministic times and no unplanned events. However in actual systems there are many sources of variation and unplanned events. For example, if resource R1 is a machine, it might break and need repair, or the actual task time for the red order on resource R5 might be 10% longer than planned.

The challenge: How to account for variation and unplanned events

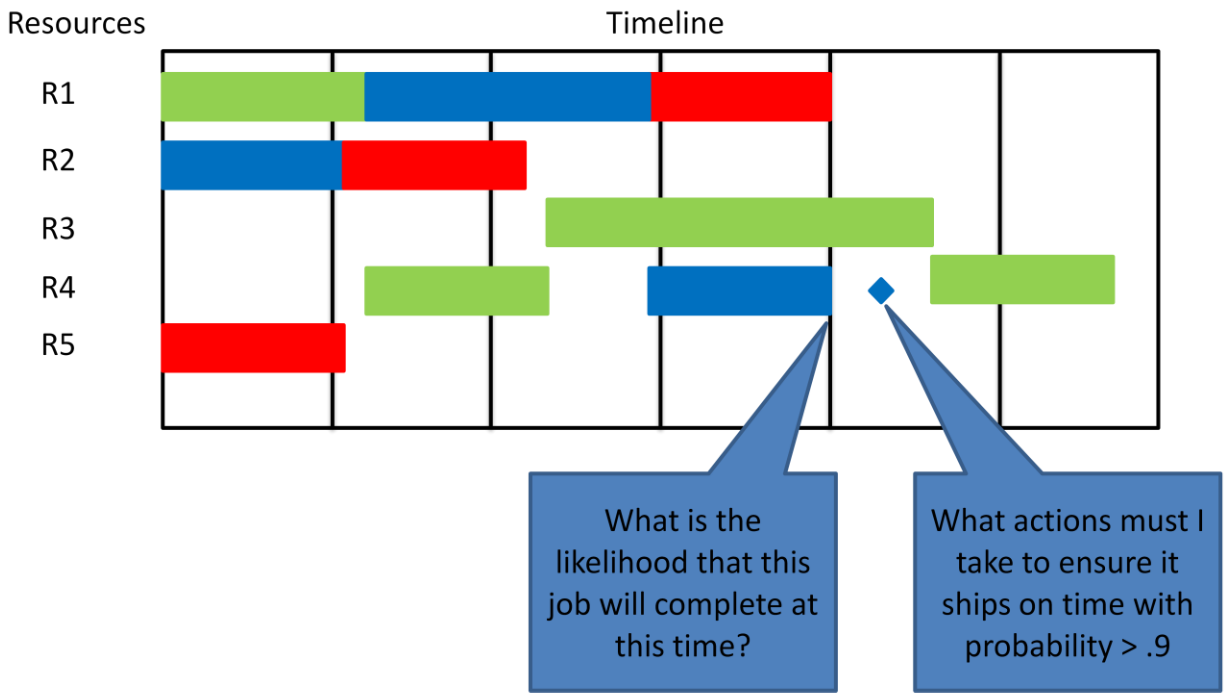
A deterministic plan is by nature optimistic, and it is rare that the plan is actually met. In typical applications the actual system performs more and more poorly over time compared to the plan, and then at some point the plan is either ignored or regenerated to reflect the variations that have occurred. It's important to realize that this does not mean that the plan was faulty; it's just the basic nature of any deterministic plan, no matter how "optimal" the plan appears from the start. A "good" deterministic plan will migrate overtime towards a "bad" plan as actual variations occur in the real system relative to the deterministic plan.

There are many sources of variation in most real systems that cause this migration from a good plan to a bad plan. Task times typically vary from their expected times, resources will often not be available as planned, and required materials may not arrive on their expected dates. In addition, machines may break, and workers may be absent or perform poorly because they are sick or distracted. Although these variations are not included when the simulation model generates a plan, they directly impact the ability of your real system to meet your plan.

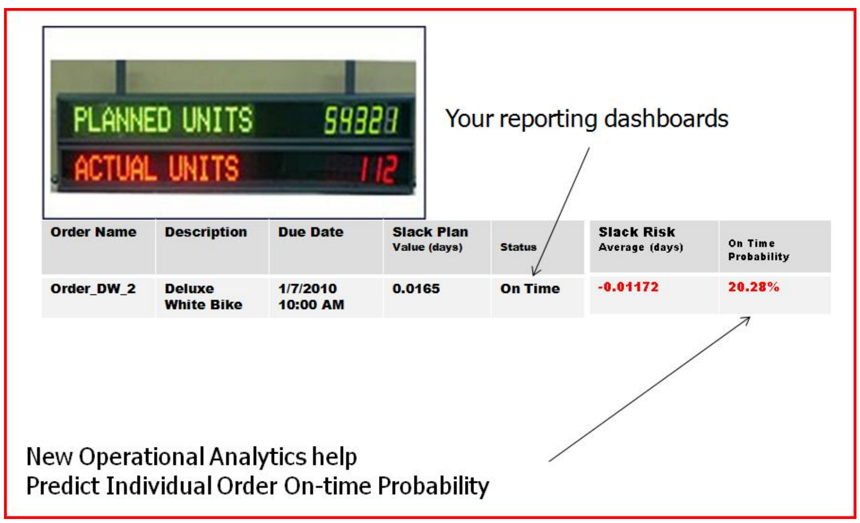
In planning and scheduling applications, you often have targets that you wish to meet for individual transactions being processed by your system. In a production system, for example, you might have targets related to delivery dates for each individual order, as well as activity-based costing assigned to each order. A feasible plan or schedule is defined as one where all targets are met by the plan/schedule. When you run the model, you generate your operational plan -- and this plan may or may not be feasible relative to the targets you have set. You can then use the model to try "what if" scenarios -- such as adding overtime, changing/splitting production batches, etc. -- to achieve a feasible plan.

However, while you may plan production in such a way that all orders ship by their due date, variation in the system may cause one or more orders to ship late. As a decision-maker, you would benefit from knowing in advance the risk associated with each transaction meeting each of the planned targets. In the previous example, you have a target ship date for your blue order (indicated by the blue diamond), and you might want to know the likelihood that the order will ship by this date.

Hence, having a feasible plan is not adequate; what you need is a feasible plan that falls within your risk tolerances for meeting your critical targets.



SOA incorporates risk measures to enable more complete evaluation



Although traditional planning and scheduling methods cannot provide any assessment of risk, with simulation-based operational analytics (SOA), you can incorporate variation and unplanned events into the same base model that you use to generate the plan to also generate risk measures for each transaction relative to its targets. As a result, a given plan can be judged not just on its feasibility at the time that the plan is generated, but also on the robustness of the plan over time in terms of the underlying risk associated with hitting each target that has been defined for each individual transaction that you are planning. This provides you with the ability to plan critical operations while fully accounting for the underlying risk imposed by variations in the system.

New simulation tools bring speed, ease-of-use and risk analysis to operational analytics

To implement SOA, a new set of simulation tools are required that focus on this general application area. Simulation tools of the past are not designed or equipped to work in this environment. These new SOA tools must support rapid modeling and easily and flexibly interface to a wide range of enterprise data that is typically held in spreadsheets, data bases, or MRP/ERP systems. These tools must also make it easy to define and properly evaluate alternatives without requiring sophisticated modeling skills or knowledge of statistics. And finally, these tools must go beyond the traditional use of simulation for comparing alternative designs, to providing information that can guide everyday production, operational, and business level decisions that are key drivers to the overall success of an assembly plant.

One such set of SOA tools, from Simio LLC, has been implemented in multiple, discrete product assembly environments. Prior to these commercial implementations, a U.S. provisional application was filed by Dr. Pegden, founder and chief executive officer of Simio LLC, and the invention has a patent pending.

# # #

About the Author: C. Dennis Pegden, Ph.D. Simio LLC founder and chief executive officer. Dennis led the development of the SLAM, SIMAN, Arena, and Simio simulation tools. He co-authored three simulation textbooks and has published papers in a number of fields including scheduling and simulation. cdpegden@simio.com.