Proceedings of the 2014 Winter Simulation Conference
A. Tolk, S. Y. Diallo, I. O. Ryzhov, L. Yilmaz, S. Buckley, and J. A. Miller, eds.

AUTOMATIC CREATION OF DAILY PSEUDO-SCHEDULES FOR A PRINTED CIRCUIT BOARD SHOP USING ARENA, ACCESS AND EXCEL

Adam Baddock, Sean Gahagan, Thomas Lhotsky, Christopher Tupino and Haiping Xu

Northrop Grumman Electronic Systems
7323 Aviation Blvd
Baltimore, MD, 21240, USA

1 INTRODUCTION

This case study describes the use of a deterministic discrete event simulation model to create a pseudo-schedule for a high-mix, low-volume printed circuit board manufacturing shop, which is managed using only queue and dispatch heuristics. By connecting a generic simulation model that emulates those heuristics to current state data extracted daily from the shop's ERP system yields a forecast or schedule of predicted events that the shop supervisor is able to use to manage human and machine resource allocation and see the impact of his decisions on predicted order completion dates.

2 PRINTED CIRCUIT BOARD FABRICATION SHOP

Printed circuit board (PCB) manufacturing is a complex business to schedule. A PCB is composed of many layers of copper-plated plastic sheets, each of which is photo-etched to create electric circuits. The sheets are bonded together, then drilled and electro-plated to create electrical connections between the layers. Such assemblies can have hundreds of operations on their process plans and they can visit the same manufacturing resources many times. The make-to-order PCB shop in this case study did no master scheduling, opting instead to control production using only queue and dispatch heuristics. However, the leadership team wished to predict delivery dates of orders in process, and to predict resource saturation.

3 SYSTEM AND PROCESS

To answer this need, we built a two-part scheduling system composed of a Microsoft Access database and a Rockwell Software Arena discrete event simulation model. The shop's ERP system makes a nightly export of current state data to an Oracle data warehouse, from which the database can query the state of the shop. Triggered by a scheduled event on a dedicated computer, the database collects data for each of the production orders currently in process. Once the data extract is completed, the simulation model is triggered to run. It reads the data in the database tables, configures the generic simulation model to reflect the current state and creates entities for each order. When the model has run to completion, the database reads the output, creates the pseudo-schedule and send the results to the shop leadership team.

4 DATABASE

The Access database is the key to the system because the data completely defines the model and therefore the schedule. Although queries which collect the current state data are complex, they are relatively straightforward. The database also contains a number of static reference tables that make the model possible. Specifically, although the ERP system contains man-readable workstation and resource codes, the simulation model can only accommodate numerical identifiers. Thus, reference tables are stored there to convert ERP data to simulation input data and also to translate the output data.

978-1-4799-7486-3/14/$31.00 ©2014 IEEE

5 SIMULATION MODEL

The Arena simulation model makes use of the Station and Resource Sets to make a flexible, generic simulation model that can be completely configured by input data. At time 0, a single entity is released into a series of data collection loops. The first loop configures the system Resources. In this model all of the Resources are modeled with a single, simple Resource Set, pre-defined with a maximum of 2200 system resources. During each pass through the loop, the entity reads a row from the resource table in the database (via the File element) and configures a corresponding member of the Set. Next, the entity populates an Array in the model with data detailing every operation that must be performed, including the Station at which it occurs, the member of the Resource Set required and the duration. A basic assumption of the simulation model is that all of the servers/workstations operate on a simple seize-delay-release principle. This allowed us to create a single-path Station Set. Our Station Set is pre-defined to accommodate up to 6000 such system stations. Next, the entity enters the final loop, where, during each pass, it spawns an order entity that is encoded with identification attributes and then routed to the queue of the Station Set member where the data table indicates it currently resides. Once the data entity completes the loops and is disposed, the model is complete and simulation time begins in earnest.

As each order entity enters a Station, it writes out an arrival message to a text file and reads operation data from the array before queuing up with other orders. Once it has advanced to the front of the queue and seized the required Resource Set member, it writes out another message to a text file before the delay begins. When the delay is complete, it writes out a third message to an external text file, then routes itself to the next Station on its process plan or, if it is complete, it routes itself to be disposed, sending one final completion message to a text file.

6 OUTPUT

Once the model has run to completion, the database compiles the schedule. The text files containing the arrival, processing, process completion and order completion messages make up the pseudo-schedule. A query in the database consolidates the messages for each operation with the original operation data and converts simulation time to real world date/time. The database then exports the data to an Excel spreadsheet and sends copies of the file to a distribution list of shop personnel. The spreadsheet shows completion dates for every operation in a simple, graphically intuitive format. However, it can easily reconfigured to show other management concerns, like resource usage by day.

7 USE

In practice, the current schedule is used as a planning tool to predict completion dates of orders in process and to forecast resource constraints. The shop leadership team departs from the pseudo-schedule as needed to accommodate changes in resource and personnel availability. Each morning, the new pseudo-schedule tells them the impact of the decision they made the previous day.

8 NEXT STEPS

Now that the team has gotten accustomed to using the deterministic pseudo-schedule, we can begin to explore more of the systems potential.

Currently, each run of the scheduling system provides feedback about the consequences of decisions made to address resource and priority changes during the previous day, but we would like to provide the leadership team an interactive way to run the model and see the impact of their decisions a priori rather than a posteriori. We intend to do this by building a better user interface for the database engine which can trigger the model to run and compile the results for different scenarios.

There is currently no uncertainty in the model. Uncertainty would add little value to the schedule, but we would like to add a set of stochastic runs to the end of the daily deterministic cycle to better characterize uncertainty in the completion dates, where deterministic results can be quite inaccurate. The initial success of the pseudo-schedule has opened the door for these enhancements that only simulation can deliver.