Proceedings of the 2005 Winter Simulation Conference
M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, eds.

ANALYZING PRINTED CIRCUIT BOARD ASSEMBLY LINES USING A PCB ASSEMBLY TEMPLATE

Pradip D. Jadhav
Jeffrey S. Smith

Industrial & Systems Engineering Department
Auburn University
Auburn University, AL 36849, U.S.A.

ABSTRACT

PCB assembly lines are characterized by asynchronous transfer of parts resulting from variability in terms of random processing, failure, repair, exhaust, and replenishment times. The throughput for such manufacturing systems depends upon simultaneous availability of resources; namely, machines, material, and operators. Further, random breakdowns along with the capacitated buffers cause blocking and starving which affects the throughput. Analytical models for such system require simplifying assumptions, hence simulation modeling is the popular choice. For simulation models to be successful, capturing the impact of operator interference is important. This paper describes a methodology developed for simulation modeling and analysis of PCB manufacturing lines, capturing the complex interactions between its components. A custom-developed PCB assembly template is used for modeling purposes. Though the analysis described in this paper pertains to PCB assembly lines, it is equally applicable to general class of serial production lines with capacitated buffers.

1 INTRODUCTION

PCB assembly lines are characterized by capacitated buffers, random processing times, production stoppages due to unreliable workstations and component parts exhaustion, and pooled operators responsible for manual operations as well as for repair and replenishment of processes. These factors make analytical models difficult to apply. One needs to make certain simplifying assumptions in order to model such systems analytically, causing concern over applicability of the model. Simulation models on the other hand, can explicitly capture these complex interactions.

In this context, simulation can be used to:

- Assess the feasibility of the process flow logic and relative impact of alternative line configurations
- Assess the ability to meet planned production rates/quantities
- Identify bottleneck operation(s) and evaluate improvement strategies
- Examine complex interaction between resources
- Identify optimal operator assignment
- Allocate buffer space .

From simulation modeling point of view, capturing the characteristics of the PCB assembly line, such as the machine failures, component part exhaustion, operators tending multiple processes, the model development becomes tedious and time consuming. Further, extracting custom reports requires additional modeling effort. Commonality of the processing steps combined with the above mentioned factors makes PCB assembly line a good candidate for development of a custom template for simulation modeling. The PCB assembly template developed by Mukkamala et al. (2003) proves extremely useful in this regard. It offers a unique simulation output category in the form of percentage of time spent by a resource in seven specific states germane to serial production system in addition to simplifying the model building process.

This paper describes a methodology developed for simulation modeling and analysis of complex PCB assembly lines. The specifics presented here pertain to PCB manufacturing lines, but the methodology is equally applicable to general class of serial production lines with adaptation of the template to suit the particular domain. The methodology involves the following steps: a) Collect input data, b) Develop a static model of process line, c) Develop and validate a discrete-event simulation model using PCB assembly template d) Run the simulation model with Current Configuration, e) Analyze the resource states for Current Configuration, f) Modify configuration and repeat.

This paper is organized as follows. Section 2 briefly describes the background for this research. Section 3 gives detailed description of each step in the six-step methodology. Section 4 presents a case study conducted using the methodology. Section 5 presents the conclusions.

2 BACKGROUND

This research is primarily based on the work described by Mukkamala et al (2003). This work focuses on developing a custom template for PCB assembly process. A typical machine on PCB assembly line includes a) an input conveyor, b) an input buffer, c) a processing station, d) an output buffer, and e) an output conveyor. The input and output buffers are optional. The machine can process boards either individually or in batches. One of the main sources of uncertainty in such manufacturing systems is unpredictable production stoppages (Goss et al. 2000). The production stoppages can be divided into two common classes: machine failures caused by irregularity in the machine, and component part exhaustion. In the PCB assembly process, various components parts (resistors, capacitors, fuses, covers, connectors, etc.) are placed onto the boards at various stages of the assembly. Inventory of these component parts is maintained on the line at corresponding stations. While the machine is in working condition (in case of automated assembly/placement operations) or the operator is available (in case of manual assembly/placement operations); if the inventory of component parts is exhausted, the production at this particular station will be stopped due to the unavailability of component parts. We categorize this type of production stoppage as *component part exhaustion*.

Once a part exhaustion occurs, a *component replenishment* is required in order to restart the process. We are interested in capturing the impact of the exhaustion and replenishment processes on the production rate. In addition, PCB assembly process involves number of inspection stages, where boards failing inspection are checked by an operator before they are discarded or sent to rework. Thus, the performance of PCB assembly line largely depends upon simultaneous availability of equipment, material, and operators. With increased automation, operators are tending groups of machines, doing some manual operations and at the same time being responsible for repairing broken machines and replenishing components parts. This can result in non-availability of operator, increasing downtime, and reducing throughput. It is important to include these complex interactions in simulation and capture the consequences of such interactions. Tracking the resource states can provide such information.

Generally simulation models are developed from scratch to solve a specific problem. With models of real life PCB lines being quite large in size, capturing the complex interactions and extracting the statistics such as resource states using modules from commercially available simulation software, makes the modeling phase very time-consuming and tedious. For this reason, Mukkamala et al. (2003) have developed a custom template for modeling PCB assembly processes. This template was developed in Arena 7.01. The following modules are present in this template: a) Board Destacker used to model the destackers with failures, b) Process Plus module used to model machines with failures and exhausts such as screen printers, placement machines, board inverters, and batch curing machines, c) Inspection module for modeling inspection machines with failures, d) Board Stacker to model the stackers with failures, e) Inline Process module to model the soldering oven, inline curing or cleaning machines with failures, f) Conveyor module for modeling conveyors with failures. These modules make the model development phase of a simulation project fast and easy. Also, use of template simplifies the verification and validation of overall model as the modules in template are already verified and validated.

As a part of the current work one more module- Multi-stage Process module was added to the template. This module is meant for modeling machines which process boards in multiple stages, and which encounter failures and component part exhaustion. For example, in placement machines like FCM, the board enters the machine on a conveyor and the conveyor is indexed to next station at a specified interval. By the time the board leaves the machine, it is fully processed, processing being done simultaneously on multiple boards.

Using the PCB assembly template enables tracking the percentage of time spent by each module in each of the following seven states:

- Busy: When the resource is doing actual processing,
- Idle: If the resource is in working condition, but, it has no boards to process,
- Blocked: After being processed at a resource, the board tries to leave the machine if there is place in the output buffer or succeeding conveyor. If there is no place in the output buffer or on the succeeding conveyor, the board remains on the resource,
- Failure: If the machine goes down, the machine waits for an operator/technician to arrive and fix the machine,
- Repair: This is the state in which the machine is undergoing the repair by the operator/technician,
- Exhaust: When one or more component parts are exhausted and the placement or assembly machine/operator can't process the boards (and is waiting for the replenishment operator),
- Replenish: This is the state of the resource when the operator is replenishing the component parts needed for the processing to start.

3 METHODOLOGY

This methodology has been developed based on the PCB assembly template and the simulation modeling and analysis projects for actual PCB assembly lines that we have worked with. The methodology involves six steps to be followed in order. These steps are explained in detail as follows:

3.1 Collect Input Data

This is one of the most important steps for successfully completing a simulation project. Simulation being a descriptive modeling technique predicts the output for some set of input conditions. More often than not, simulation modelers lack the domain specific knowledge, and have to depend upon the decision makers of the particular domain (Mukkamala et al 2003). So, with most real systems being complex and stochastic in nature, it is of prime importance to ensure that the input data being plugged into simulation model represents the actual systems fairly accurately. For this purpose the simulation modelers and the decision makers pertaining to the particular domain should work closely to extract accurate data from the system.

This data may be extracted from historical databases of the line under consideration (if the manufacturing line already exists) or a similar line (if line is non-existent, i.e. proposed). For making the data collection step fast and easy we have standardized the set of input parameters, which are summarized as follows:

1. Machine Data: If a machine is batch processing or individual processing, cycle time for the process; failure data such as distribution for "Time to Failure"/"Cycles to Failure", distribution for "Time to Repair", and repair resource; exhaust data such as distribution for "Time to Exhaust"/"Cycles to Exhaust", "Time for Replenishment", and replenishment resource
2. Operator data: Tasks allocated, and task times
3. Inspection and rework data: Passing percentages at inspection stages, rework resource, and time for rework
4. Oven data: If it is an inline process, the length and speed of conveyor are required; or if it is a batch type process, the batch size and delay are required
5. Buffer data: Capacity of the buffer, and the queuing discipline (e.g. LIFO/FIFO, etc.)
6. Conveyor data: Length in terms of number of parts that can be accommodated, and speed of the conveyor
7. Transporter/shuttle data: Loading time, transfer time, and unloading time
8. Layout of the line depicting the detailed process flow logic.

3.2 Develop Static Model for the Process Line

Based on the input data collected in Step 1, a static model is prepared for the validation and benchmarking of the simulation model. In the static model, expected production/throughput is calculated for each individual process using the mean values cycle times, failure/repair rates, and exhaust/replenishment rates. Then, the production/throughput which is the least among all the machines/processes is identified as the production/throughput of the whole line. In doing so we ignore the inherent interdependence between the processes (induced by the capacitated buffers), the variability (induced by the unpredictable breakdowns and component part exhaustions), and the reentrant flow after detection of failures at various inspection stages and subsequent rework.

The input parameters like the available production time, the cycle times, failure/repair rate, and exhaust/replenishment rates can be changed easily to adapt to updates in the values of the same. Three scenarios are developed for the static model as follows:

1. Scenario 1: Ignoring both machine failures and parts exhaustion
2. Scenario 2: Including machine failures but excluding parts exhaustion
3. Scenario 3: Including both machine failures and parts exhaustion

The production quantities thus obtained are the upper bounds on the actual production. At this stage, the decision makers and domain experts of the particular domain should be consulted to verify that these upper bounds obtained are representative of the reality. Thus, in effect static model also works as a check on the input data to be used for simulation. Once the static model is verified with the decision makers and domain experts, it is ready to be used for the validation and benchmarking of the simulation model.

3.3 Develop and Validate the Simulation Model

The simulation model differs from static model by explicitly considering capacitated buffers between each pair of processes, operator interference when operators are responsible for manual operations as well as for repair and replenishment of multiple processes, variability in terms of cycle time, failure/repair rates, and exhaust/replenishment rates, detection of failures at various inspection stages and subsequent rework. The PCB assembly template was originally developed by Mukkamala et al. (2003) using Arena 7.01. The simulation models for this study are also developed in Arena 7.01 using this custom template as well as Arena built-in templates.

The model developed using the PCB assembly template is first verified by using an animation run. Then it is validated by comparing it with the static model developed in Step 2. The configuration modeled for validation is called the Base Configuration. It is modeled in line with Scenario 1 of the static model, where machine failures and part exhaustions are ignored, there is no operator interference (each process/task assigned a separate operator, so that there is no time lost in waiting for an operator), detection of failures at inspection stages and further rework is

ignored, mean values are used for cycle times (these are usually deterministic for PCB assembly line).

3.4 Run the Simulation Model with Current Configuration

Current configuration resembles the design as proposed by the manufacturer. It incorporates variability in the processing times, failure/repair rates, exhaust/replenishment rates. The detection of failures at inspection stages and the subsequent reworks are modeled here. Also, it incorporates the operator assignment to the multiple processes as proposed by the manufacturer. A trial run is made with the current configuration. As current configuration involves sources of randomness, after observing the half-width of the confidence interval of the output statistic (usually the throughput) the number of replications to be run is decided. The output statistic will then predict the ability to meet the planned production. Then, a resource state graph is prepared based on the statistics provided by the PCB assembly template modules. The resource state graph and its analysis to come up with improvement strategies is explained in next section.

3.5 Develop and Analyze the Resource State Graph

The ability to distinguish the resource states can provide great insight into the operational details of the manufacturing system. The Busy state essentially tells the effective utilization of the resource. Idle and Blocked states signify the impact of starving and blocking caused by the inherent speed mismatch, machine failures, and component part exhaustions. Repair and Replenish states show the productive time lost due to machine failures and component part exhaustion. Failure and Exhaust states capture the complex interactions between the resources, namely, machines and operators. These two states combined represent what we call as "operator interference". This is the time lost due to unavailability of the operators when the machine goes down. In case of the PCB assembly lines, which is characterized by highly automated machines, each operator monitors a group of machines (this group usually comprises five to six machines). The operator is responsible for attending the machine failures and component part exhaustions. If the line is understaffed or if tasks aren't allocated correctly, there could be many instances when one of the resources can't continue processing, and waits for the operator to fix the machine (or replenish the component parts) as the same operator is tending some other machine. Thus, operator interference could have great impact on throughput of the line. Moreover, the production loss due to interference is not reflected in the static model.

The term "Machine interference" can also be found in the literature. Stecke (1985) defines machine interference as the time that the machine is available to run but is waiting for an operator to finish tending other equipment. It only considers the time when both machine and material are available simultaneously, but, the machine is not processing. This is the distinguishing factor between what we call operator interference and machine interference. Machine interference does not take into account the time spent waiting for the operator for repair/replenishment when one of the resources can't continue processing because of machine failures or component part exhaustion. The literature considers this waiting for operator when machine can't process as "Down" state (which also includes the actual repair time), whereas we regard this state as either Failure or Exhaust (depending upon whether machine encountered a failure or component part exhaustion), and these two states combined being regarded as the operator interference. Machine interference could be more appropriate for situations where significant time is spent in setups. In case of PCB assembly lines, one of the most important factors decreasing productivity is the unpredictable downtime. Hence, we stick with operator interference which helps to concentrate on the waiting for the operator when the resource can't continue processing due to either machine failure or component part exhaustion.

In the Resource State Graph, all the resources on the line are laid across the X-axis in the same sequence as the they appear on the line. The percentage of time spent by each resource in each of the seven states is stacked along the Y-axis in the form of bars. All the states are color coded. A sample Resource State Graph is shown in Figure 1.

For analyzing the Resource State Graph, we look for specific patterns. For example, in Figure 1, the resources upstream of Load Component 2 are blocked for a significant amount of time (indicated in yellow), whereas the resources downstream to Load Component 2 are starved most of the time. From the above observations it is pretty clear that Load Component 2 is the bottleneck for this scenario. Once we find this out, we can seek the reason for its being the bottleneck, and in this case it's the time spent in Replenish state. We can also see whether there is significant amount of operator interference. The operator interference is essentially represented by the Failure and Exhaust states (shown in sky blue and orange respectively). If a set of machines encounters a significant amount of operator interference (i.e. it spends significant time in Failure or Exhaust or both states), then one can conclude that this particular set of machines is understaffed, and some sort of operator reallocation is required. In case of Figure 1, the impact of operator interference is negligible (indicating that the line is either overstaffed or it is staffed optimally).

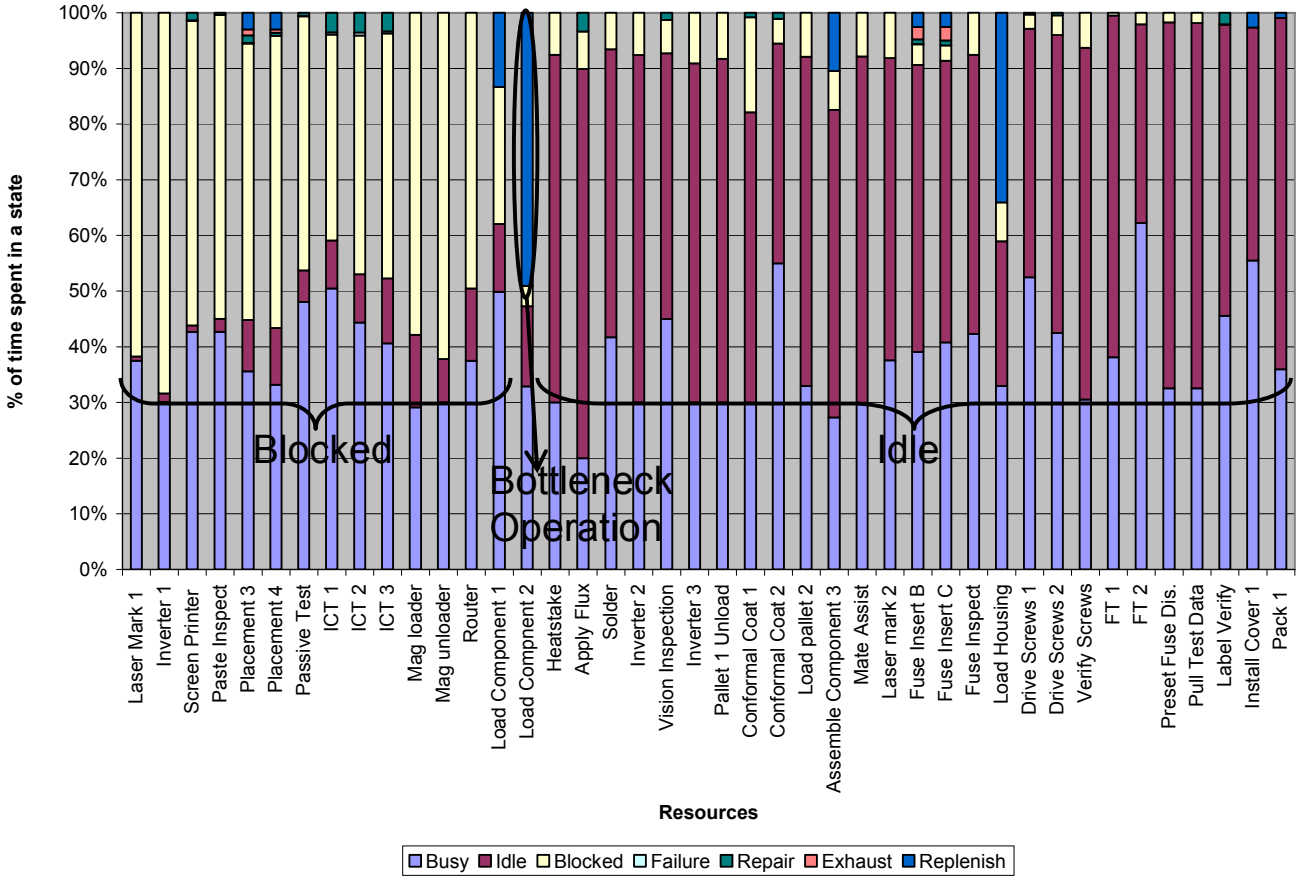

Figure 1: Sample Resource State Graph

In such situation one can try to reallocate the operator tasks decreasing the number of operators and see if it affects the throughput. Further, by looking at the time spent in Idle and Blocked state one can try to reallocate the buffers.

For example, the bottleneck in Figure 1 (resource Load Component 2) spends around 10% of time in Idle state (this is because of dependence on the upstream processes, induced by the capacitated buffers). One can try increasing the size of buffer before resource Load Component 2 to reduce severity of this bottleneck resource (provided that there is no other simpler way to do it) and see the impact of doing so. Thus, the Resource State Graph provides an excellent visual tool to analyze the production line and help develop improvement strategies.

3.6 Modify Configuration and Repeat

The Resource State graph is analyzed as explained in the previous section. Based on this analysis, a set of improvement strategies is developed. The decision makers are consulted on these improvement strategies and the most feasible one is selected for further consideration. The configuration of the simulation model is modified accordingly. Then, the simulation is run with this modified configuration. The results are then discussed with the decision makers and decision is made if some other improvement strategy needs to be considered and tested.

Once this process is over, the Resource State Graph is prepared for the modified configuration. This Resource State Graph is then analyzed to identify the new bottleneck or potential for further improvement. The improvement strategies are developed accordingly.

Then, the Steps 5 and 6 are repeated iteratively until there are any significant discernable patterns shown by the Resource State Graph which could lead one to some improvement strategy.

4 CASE STUDY

The methodology developed was used for simulation modeling analysis of a real PCB assembly line for a local PCB manufacturer. At the start of the study, the manufacturer had an initial design for the line and was interested in determining whether the configuration would meet planned production requirements and in identifying ways

to improve line productivity. We will refer to the line as Product x line. Note that the real data has been coded to protect proprietary information.

The process started with the Input Data collection step. As Product x line is proposed (non-existent), historical data from other similar lines (which have many processes in common) was collected in the format as mentioned in Section 2.1. The failure data was collected from the machine logs; the data on stockouts was calculated from the planned inventory of the component parts to be available at the machines. Three different board variants will be processed on this line with the production mix as follows: Variant A-34%, Variant B-39.5%, Variant C-26.5%, with a changeover time of 10 minutes between different variants. It is assumed that bare boards are always available for production. The total production time available is 347,616 minutes (after deducting breaks).

Based on the input data collected the Static Model was developed and the total throughput (including all three variants) as given by the Static Model is depicted in Table 1 below:

Table 1: Static Model for Product x Line

Scenario	Scenario 1	Scenario 2	Scenario 3
Expected Boards per year	931,719	931,337	700,486

After verifying the results of the Static Model with the manufacturer, simulation model was developed and two configurations inline with the Scenario 1 and Scenario 3 were run. The results are summarized in Table 2.

Table 2: Comparison of Static Model and Simulation Model for Product x Line

Scenario	Boards Produced per Year	
	Static Model	Simulation Model
Scenario 1	931,719	931,700
Scenario 3	700,486	523,408

For the Base Configuration (which resembles Scenario 1), the results of simulation model and the static model match closely, and hence we have some strong evidence that the simulation model is valid. The difference between the numbers for static model and simulation model for Scenario 3 (which is the Current Configuration) is due to: the operator interference, variability in the failure and exhaust data, and the detection of board failures ignored at various inspection stages and subsequent rework, which are not considered by the static model. Despite the difference, the comparison verifies that the boards produced for Scenario 3 are within the upper bound set by the static model.

After validating the simulation model, the Resource State Graph is prepared based on the resource state statistics provided by the modules from PCB assembly template . The Resource State Graph for the Current Configuration of Product x Line is shown in Figure 1. As indicated in Figure 1, we can observe the pattern in which the resources upstream to Load Component 2 are blocked for significant amount of time whereas the resources downstream are idle for most of the time. Hence, Load Component 2 was identified as the bottleneck for the Current Configuration. This is a manual operation, and as observed from Figure 1, the operator spends approximately 50% of the time replenishing Component 2.

We distinguish the inventory of component parts as either on-hand or off-hand inventory. On-hand inventory is the one available in small totes at the workstation, and off-hand inventory is the one which is available away from the machine in larger containers. Three improvement strategies are possible here: a) Reduce the time required for replenishment of Component 2 container, b) Assign Component 2 replenishment task to some other operator, c) increase the on-hand and off-hand inventory available for Load Component2. Out of these three alternatives, the alternative of assigning the task of replenishing the off-hand inventory to a material handler was chosen after consultation with the manufacturer. The current configuration was modified to incorporate the change. We refer to this configuration as the Alternative Configuration 1.

A simulation was run with this modified configuration. For the Alternative Configuration 1 the throughput increased to 676,257- an increase of 29.2%. The Resource State Graph for this configuration is shown in Figure 2. This Resource State Graph was then analyzed to assess the potential for further improvement. By observing Figure 2, it can be observed that the blocking of the processes at the start of the line and starving of the processes at the end of line has decreased to certain extent. But the discernable pattern of blocking of a group of processes and starving of the rest of the processes could still be observed. The processes upstream to Load Component 1 are blocked for significant amount of time and processes downstream to it are significantly starved. This indicates that the bottleneck has now shifted to the process Load Component 1. The on-hand inventory of Component 1 was increased from 12 to 35 and off-hand inventory was increased from 180 to 280. This is referred to as Alternative Configuration 2. The throughput increased to 738,035, an increase of 9.13% over Alternative Configuration 1. The Resource State Graph for this configuration is shown in Figure 3. The utilizations of the resources (indicated by Busy state) has gone up by certain amount

It can be observed from Figure 3 that no discernable pattern, as was observed in case of Current Configuration and Alternative Configuration 1, is present. The processes Conformal Coat and Install Cover have the highest utilizations (which is expected as their cycle times are on the

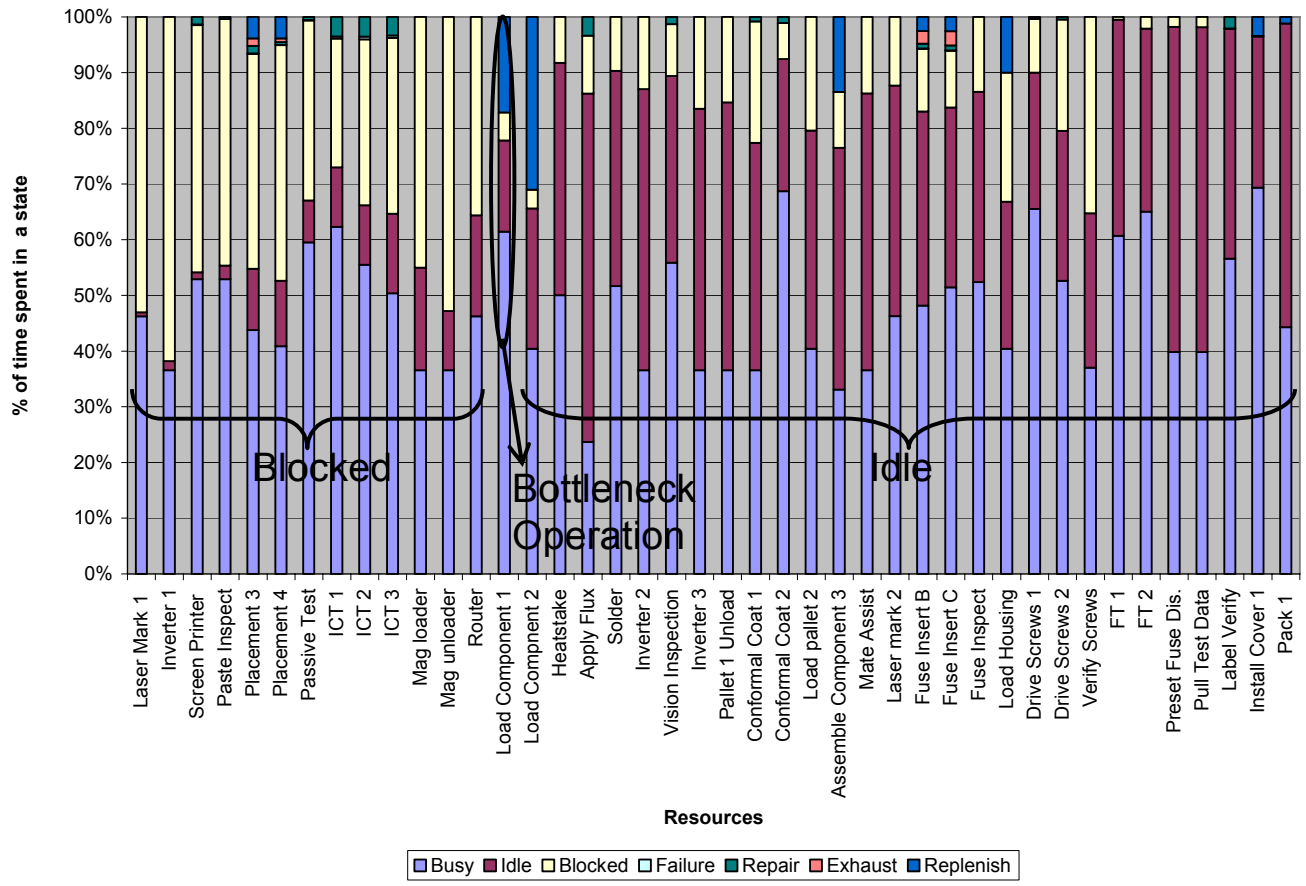

Figure 2: Resource State Graph for Alternative Configuration 1

higher side). So, no single process could be targeted for further improvement without altering the cycle time. Also, no significant operator interaction (indicated by time spent either Failure or Exhaust state) could be observed, indicating that the line is not understaffed. Buffer is provided in the form of magazines which are transported through Magazine Loader and Magazine Unloader. Additional experimentation was done by increasing the number of magazines present at Magazine Loader and Unloader. This can decrease the blocking of processes upstream to Magazine Loader and starving of the processes downstream of Magazine Unloader. The increase in production with increasing the number of magazines is shown in Table 3.

Thus, the case study for the Product x Line demonstrates how resource state statistics provided by modules from PCB assembly template can be used in the form of a Resource State Graph to develop improvement strategies. It was shown how the improvement strategies could increase the production from 523,408 to 783,603, an increase of around 49%.

Table 3: Effect of Increasing Number of Magazines on Throughput

Number of Magazines	Number of Boards Produced
1	623,611
4	738,035
12	769,298
20	783,603

5 CONCLUSIONS

The modules from the PCB assembly template simplify the model development and reduce the modeling efforts required to incorporate the machine failures, part exhausts, and to extract the statistics like the resource states. The importance of ability provided by the template to distinguish seven different resource states is discussed. A six-step methodology is proposed for simulation modeling and analysis of PCB assembly lines. The development and analysis of the resource state graph is explained and its use in identifying problem areas and developing improvement strategies is demonstrated with the help of case study for Product x line. Proper application of this methodology can result in developing improvement

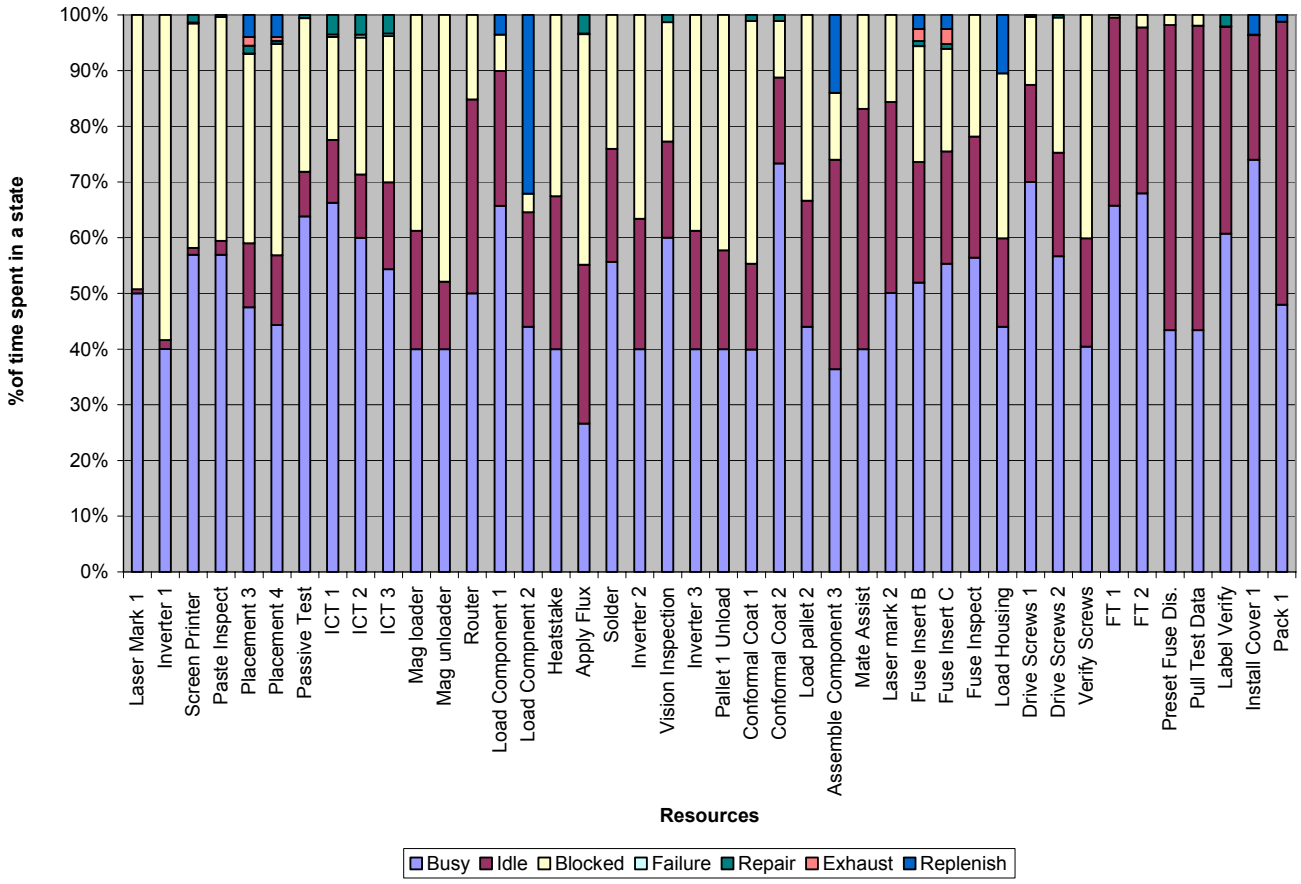

Figure 3: Resource State Graph for Alternative Configuration 2

strategies which could fetch significant improvements in terms of throughput and operator allocation.

Further extension of the proposed methodology is underway, which involves automating the process of developing simulation model from the static model. Also, efforts are being directed to assess the potential of formalizing and automating the process of analysis of resource state graph and development of improvement strategies based on this analysis.

REFERENCES

Goss, B. G. 2000. Measuring machine interference to evaluate an operator cross-training program, Sloan School of Management and Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts.

Mukkamala, P. S., Smith, J. S., and Valenzuela J. F. 2003. Designing reusable simulation modules for electronics manufacturing systems. In *Proceedings of the 2003 Winter Simulation Conference,* eds. S. Chick, P. J. Sanchez, D. Ferrin, and D. J. Morrice, 1281-1289

Stecke, K. E., and Aronson, J. E., 1985. Review of operator/machine interference models. *International Journal of Production Research* 23-1: 129-151

AUTHOR BIOGRAPHIES

PRADIP D. JADHAV is an M.S. student in the Department of Industrial and Systems Engineering at Auburn University. He received his B.S. degree in Mechanical Engineering from D. Y. Patil College of Engineering and Technology, Shivaji University, Kolhapur, India, in 2000. His research interests include simulation modeling and analysis of manufacturing systems. His e-mail address is pradip_jadhav@yahoo.com.

JEFFREY S. SMITH is a Professor in the Industrial and Systems Engineering Department at Auburn University. Dr. Smith received the Ph.D. and M.S. degrees in Industrial Engineering from Penn State University in 1992 and 1990, respectively, and the B.S. in Industrial Engineering from Auburn University in 1986. His research and teaching interests include applications of simulation and simulation language design and manufacturing systems design and analysis. Dr. Smith's email address is jsmith@auburn.edu.