Discrete Event Simulation for Supporting Production Planning and Scheduling Decision in Job Shop Facilities

Ashton Allen, Jesse Caldwell, Christopher Heard, Ioannis Sakiotis, Daniel Tillinghast
Department of Modeling, Simulation and Visualization Engineering, Old Dominion University
Norfolk, Virginia 23529
[Aalle041, Jcald013, Chear008, Isaki001, Ddrake006]@odu.edu

ABSTRACT

This paper describes a new simulation-based production planning and scheduling (PPS) system for use in managing a job shop manufacturing facility. A job consists of a process flow that contains all tasks that must be accomplished in order for the job to be complete. Each task can be performed in several workstations (resources) of varying performance. The order in which the task will be performed, as well as the resources that will be involved in the process is called a schedule and it is specified by the shop managers. However, when creating a new schedule, it is difficult to predict and account for the randomness in the system as well as the contention for facility resources that is likely to occur. The PPS system will accept a proposed job schedule and, by using simulation, will resolve the resource contention. It will produce a realizable job schedule that allows the user to see the impact of resource contention on the shop floor. This paper describes the high-level design of the PPS system, with an emphasis on the input and output sub-systems.

ABOUT THE AUTHORS

Ashton Allen is a senior at Old Dominion University. He is majoring in Modeling and Simulation Engineering and minoring in Computer Science. Ashton is planning on completing the accelerated Master of Engineering program in Modeling and Simulation. He has accepted a job offer with Newport News Shipbuilding and is going to be working with their Modeling and Simulation department after graduating with his Bachelor's in May 2014. His research interests are serious gaming, discrete event simulation, video game design and development, and statistical methods.

Jesse Caldwell is a senior in the Modeling and Simulation Engineering program at ODU. He is interested in the development of serious games and simulation applications. Jesse is looking forward to working in the industry upon graduating in May 2014.

Chris Heard is a senior in the Modeling and Simulation Engineering degree program at Old Dominion University. He is graduating in May 2014, with his Bachelor of Science. He is planning to continue his education by pursuing a Master's degree program while working full time. He would like to find a job focused in modeling and simulation in the field of meteorology.

Daniel Tillinghast has spent several years in the U.S. Navy as a cryptologic technician before becoming fascinated by "serious gaming" and "cognitive modeling" and pursued his education in Modeling and Simulation Engineering. During the course of his education, he has developed and executed simulations exercises which have applications in commerce, virology, biology, defense and manufacturing. With this wide range of practical experience, he is pursuing the development of his education which will allow him to apply these skills in the marketplace after his graduation.

Ioannis Sakiotis is a senior in the Modeling and Simulation Engineering degree of Old Dominion University and is minoring in Computer science. He is expected to graduate in the summer of 2014. Upon graduating he is planning on pursuing a Master's Degree in Modeling and Simulation. His research interests are autonomous robotic systems, transportation, serious gaming and military applications of modeling and simulation.

Discrete Event Simulation for Supporting Production Planning and Scheduling Decision in Job Shop Facilities

Ashton Allen, Jesse Caldwell, Christopher Heard, Ioannis Sakiotis, Daniel Tillinghast
Department of Modeling, Simulation and Visualization Engineering, Old Dominion University
Norfolk, Virginia 23529
[Aalle041, Jcald013, Chear008, Isaki001, Ddrake006]@odu.edu

INTRODUCTION

The Senior Capstone Design Team (SCDT) from the Department of Modeling, Simulation, and Visualization Engineering (MSVE) at Old Dominion University (ODU) is creating a prototype of a simulation-based, adaptive production planning and scheduling (PPS) system. This prototype is being designed and implemented for Newport News Industrial (NNI) with support from the Newport News Shipbuilding (NNS) Modeling and Simulation department. The SCDT is participating in a two semester capstone design course which is mandatory for students majoring in Modeling and Simulation Engineering at ODU. In this course, the students work with an outside customer on a project to prepare the students for the transition from the university to the professional workforce. The students utilize the engineering design process to define the problem, design a solution, and develop a prototype implementation. The 2014 graduating class is working with NNI and NNS to complete the Capstone Design course.

This paper describes the design of a PPS system that will assist NNI with their production planning and scheduling process. The PPS system will be used to create, edit, or evaluate schedules and aid in the decision-making process of accepting new work. Each job consists of a sequence of tasks. Each task requires one or more facility resources in order to be processed. All of the jobs in the facility are competing for resources, which is leading to resource contention. In addition, certain tasks may require rework when a job fails an inspection due to a defect or unsatisfactory work. The possibility of rework and resource contention makes creating accurate job schedules difficult. The purpose of the proposed PPS system is to assist in developing a realizable schedule.

This paper is divided into seven sections. In the Problem Definition section, a brief description of the customer and their current planning process is provided. The Solution Approach section presents the SCDT's conceptual design of the PPS system. The System Overview section expands on the conceptual design by describing the three sub-systems, the System Input, System Output, and PPS Simulation. The Input System section describes the input requirements and graphical user interfaces (GUI) for the PPS system. The Output System section describes the output data that is displayed. Finally, the conclusion describes the current status of the project and possible additions for future development.

PROBLEM DEFINITON

Newport News Industrial (NNI) is a subsidiary of Huntington Ingalls Industries. NNI provides a wide range of services including fabrication, construction, equipment repair, and technical services. NNI's Oyster Point Industrial Park (NNI-OP) facility is the main customer of this project. Their core capabilities consist of machining, welding, rigging and mechanical assembly. NNI-OP is not a traditional assembly line manufacturing plant. This facility processes a variety of complex jobs, each with a unique process flow. The variation between jobs makes scheduling difficult, especially when deciding to accept new jobs. When deciding to accept new jobs, the facility planners must determine the process flow for the new job, including the required resources and task durations. Then they must determine the impact the new job has by considering the possible effects the new job can have on jobs currently in the facility.

Currently, NNI-OP generates a document, known as a Job Tracking Sheet (JTS), to document the tasks that define a job and all their requirements. The JTS separates a job into tasks, which represent different processes that must be finished in order for the job to be completed. Each task specifies the resources required and the task's processing time.

The JTS is excellent for organizing jobs into tasks and displaying vital information relevant to the job. However, the JTS does not take into consideration resource contention throughout the facility and the probability of rework. Therefore, the time a task spends completing rework or waiting for a resource cannot be accurately accounted for on the JTS. However, if the resource contention and rework times could be accurately modeled, then the planner could create a more realizable schedule. In an attempt to assist the planner, this project aims to create a PPS system that will give the planner the ability to evaluate the impact of rework and resource contention on the facility.

SOLUTION APPROACH

In order to assist NNI with the production planning and scheduling process, the development of a PPS system is proposed. The purpose of the PPS system is to make the schedule evaluation process faster and more accurate. The PPS system does so by simulating a collection of jobs through the NNI-OP facility and provides results about the jobs in the system and the facility resources.

A proposed job schedule, called a scenario, and the facility resources will serve as the input for the PPS system. The scenario will then be simulated, resolving job rework and resource contention. Upon the completion of a simulation run, a set of Gantt charts will be produced that display the job and resource schedules. More statistical data about the scenario performance can be displayed by doing multiple simulation runs.

The results help the user evaluate the effectiveness of implementing this scenario in the facility. Using the simulation results, the user can identify problems in a job schedule (resource underutilization, large waiting times, number of jobs that were not completed in time, etc.) and make changes to the job schedule to resolve these issues. This gives the user the ability to make educated decisions about the job schedule and continue altering the job schedule until a satisfactory result is found. By utilizing the PPS system, the user gains access to quantitative information about resource contention, variation in task durations, and rework, which are difficult to obtain without simulation [1].

It is important to note that the PPS system will not provide an optimal schedule. In order to do so, the scenario would have to be run until all possible parameter values have been tested and all possible job schedules have been generated. This would require too many simulation runs in order to get every possible job schedule, which would be extremely inefficient and would classify this as an NP-hard problem [2]. Instead the PPS system will provide information to help the user create a realizable schedule, through simulation parameter experimentation.

SYSTEM OVERVIEW

The PPS system has three main sub-systems as seen in Figure 1: the System Input, the PPS Simulation, and the System Output. The System Input sub-system receives data from the user, including job task flows, task duration times, and facility resources. The information is stored in a format that the PPS Simulation sub-system can read. In turn, the PPS simulation receives the data from the System Input and generates the job task flows and a model of the facility resources. It then runs the simulation, simulating rework and resource contention. Once the simulation run is completed, the simulation outputs raw simulation data into the Output Repository. Finally, the System Output sub-system analyzes and displays the data stored in the Output Repository; it does so by separating single run (Gantt charts) and multiple run results (statistics). Each sub-system is independent and interacts with the other sub-systems through the defined interfaces.

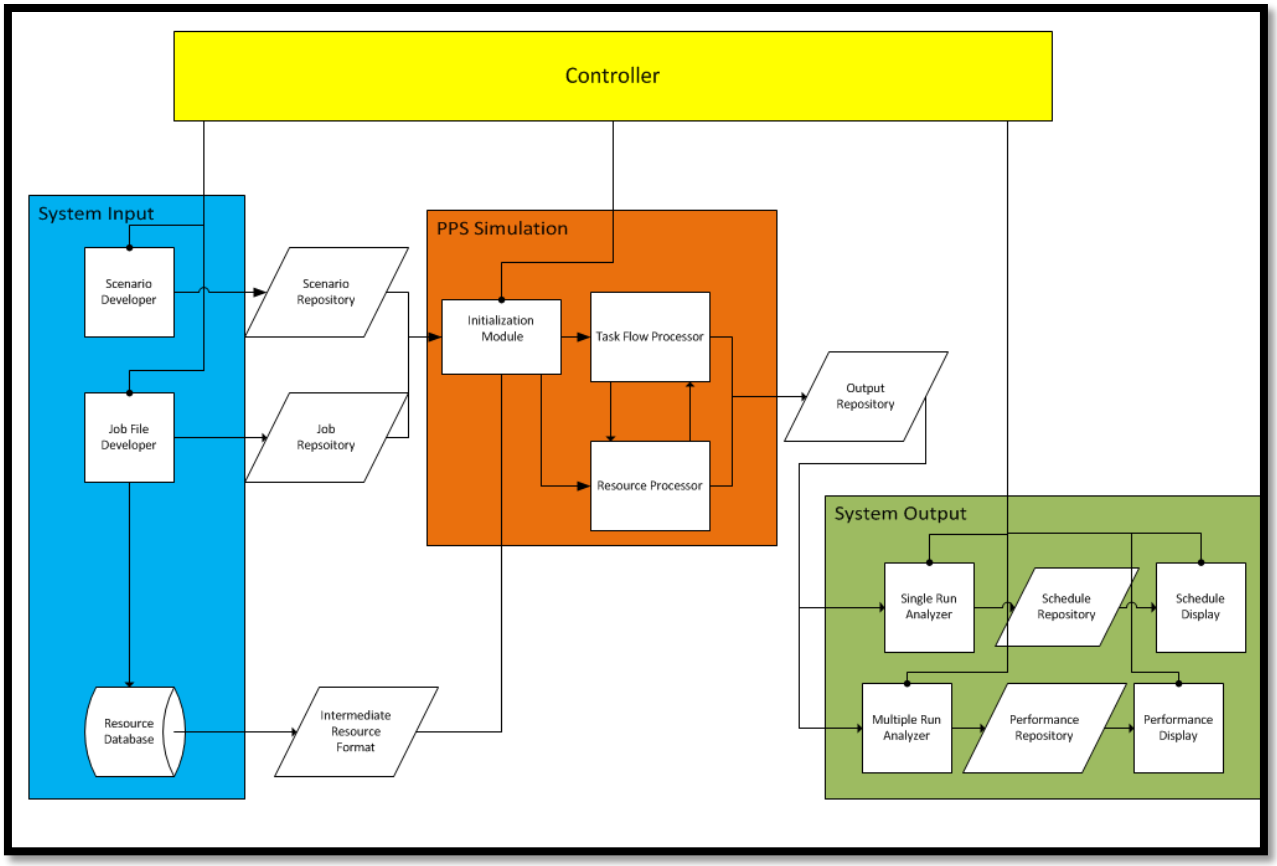


Figure 1. System Architecture

System Input

The system input sub-system has three components: the Job Developer, the Scenario Developer and the Resource Database. The Job Developer allows the user to input information about a job, including all possible tasks, their duration, and their required resources. A drop down menu with the available resources will be provided in order to prevent the user from selecting resources that are not present in the Resource Database. Besides creating job files, the user will have the ability to edit an existing job file or use it as a basis for a new job file. Once a job file has been created or edited, it will be converted to a specific job file format and stored in the Job Repository. The Scenario Developer gives the user the ability to select a collection of jobs, job start dates, priority status, and inter-arrival times, all of which define a scenario file. The jobs that are selected *must* be defined in the Job Repository prior to creating the scenario file. Once the scenario file is created, it will be stored in the Scenario Repository. The scenario itself is accessed by the PPS simulation. Finally, the Resource Database contains information about all the resources in the facility, which is saved as a resource file that the PPS simulation sub-system can read.

PPS Simulation

The PPS simulation sub-system has three components: the Initialization Module, the Task Flow Processor, and the Resource Processor. The Initialization Module reads the scenario file, generates the job task flows, and the facility resource model. The Task Flow Processor manages the task flows of the jobs in the system and sends resource requests to the Resource Processor during the run. When resources are requested, the Resource Processor provides them to the tasks in priority order, thus resolving resource contention.

System Output

The output sub-system has four components: the Single Run Analyzer, Multiple Run Analyzer, Performance Display, and Schedule Display. The Single Run Analyzer takes raw data from a single run from the Output Repository, massages the data, and stores the result into a specific file format the Schedule Display can read from the Schedule Repository. The Schedule Display accesses the Schedule Repository data and generates Gantt charts of the jobs and resources in the scenario. The Multiple Run Analyzer reads data from multiple simulation runs. It analyzes several scenario runs and computes statistical estimators for variables such as job completion times, average time in queue, and resource utilization. After the statistical estimators have been computed, they are converted into a specified format that the Performance Display can read. The statistical estimators are stored in the Performance Repository. Then, the Performance Display displays the statistical estimators in a table.

INPUT SYSTEM

The Job Developer's purpose is to assist the user in inputting the required information about a job. Besides entering job information, the user can edit job files or use existing job files as a template for a new job file. In order to make this process easy for NNI-OP, the GUI is designed to resemble an NNI-OP JTS. A job consists of all possible tasks and sub-jobs that need to occur and all the resources that are required to process the tasks. Each task in the job must have at least one resource associated with it and an estimated time duration for each resource. Some tasks can use one of multiple machines, each having their own estimated time duration. The order that the tasks occur is required, including information about all possible paths a task can follow.

A completed Job Developer GUI can be seen in Figure 2. Upon opening the Job Developer GUI, the sheet is empty. The user can insert a task by clicking the "Add Task" button on the upper left corner. This creates a new row with an empty textbox for the task name and two buttons, labeled "Resource" and "Next". These two buttons allow the user to select the resources suitable for the task and define its position in the task flow. Clicking the "Resource" button will create a pop-up box, where the user can select the resources that are suitable for the task. Some tasks require no facility resources. An option for "No Resource" is provided. Figure 3 shows an example of the Resource GUI, here the user will select a resource from the Resource Database and select a distribution and its parameters to describe the task duration.

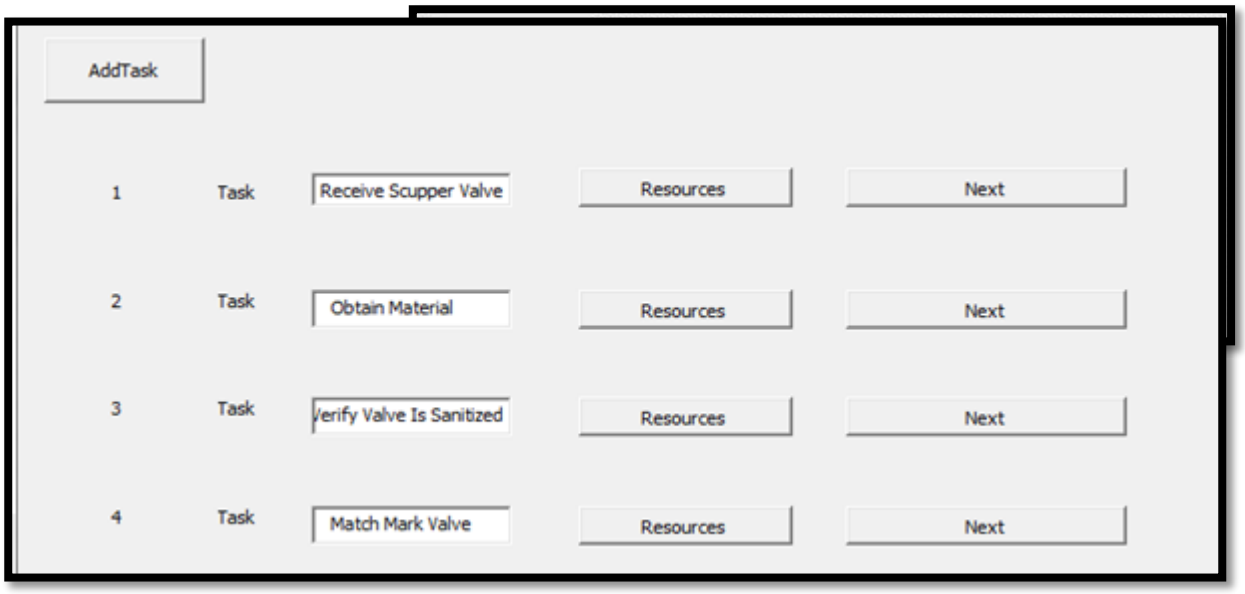


Figure 2. Job File Developer GUI

The "Next" button, in the Job Developer GUI, allows the user to define the position of each task in the task flow. This is accomplished by providing an identifier (task number) for the next task(s). The SCDT has identified three types of

tasks, a regular task, a dependent task, and an independent task, which captures the different task flow possibilities. A regular task is part of a linear task flow and is simply the next task in line. Of course not all tasks are linear, some tasks can have more than one following task that depends on the results of an inspection or a test. There are two types of tasks that meet this requirement, dependent tasks and independent tasks. Dependent tasks have two or more following tasks whose probability of occurring *must* add up to one. Independent tasks have two or more following tasks whose probability does not have to add up to one. An example of a dependent task is a "test" task, where the test can either pass or fail. This task would have a probability assigned to the following task when it passes and for when it fails. An example of an independent task is a "work inspection" task, where a job is being inspected to determine what work needs to occur.

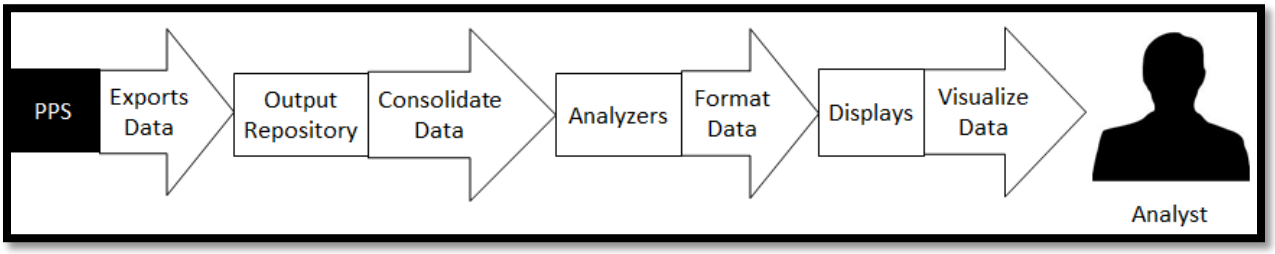
A task consists of a type, name, and a list of resources with their processing times. A job can contain a task that spawns another job, this is known as a sub-job. An example of a sub-job is when an inspection leads to a new job that is otherwise not planned for. When adding a new task, the user will have the ability to classify it as sub-job if necessary, in the Job Developer GUI. The process for inserting information for a task and a sub-job is identical. However, the sub-job must be defined in the Job Repository to be considered valid. Upon selecting the sub-job name text box, an empty Job Developer sheet will be displayed. Here the user can complete the sub-job definition by defining the tasks as described in this section.

Scenario Developer

The Scenario Developer's purpose is to create a scenario, which is a collection of defined jobs. The format of the Scenario Developer resembles that of the Job Developer, where rows are inserted on a blank sheet by clicking the "Add Job" button and filling in the text boxes and button fields. An important difference is that the user defined the tasks as they created the job file in the Job Developer, whereas in the Scenario Developer, the user can only select pre-defined jobs. This is accomplished via a drop-down menu that shows all the jobs present in the Job Repository. When adding a Job row in the Scenario Developer sheet the "Start Time" and "Inter-arrival time" text boxes will appear. There, the start time is inserted by typing a date and time. The inter-arrival box is used to specify the arrival distribution of a job. Finally, the row will also include a drop down "Priority" box. The priority can have two values: normal and high. Tasks within a job of high priority will bypass tasks with normal priority when they are contending for resources. If two tasks have the same priority, the task that has arrived in the queue first will be processed first.

OUTPUT SYSTEM

Once a scenario run has completed, the data is exported into the Output Repository. The data is then read into either the Single Run or Multiple Run analyzer for analysis. The analyzers convert the data into a format which the displays can read. Finally, the displays visualize the information needed by the user to determine the scenario's performance. Figure 4 shows a flow chart of scenario data as it moves from the Output Repository to the Displays.



Output Data

The analysis of a scenario's data from the PPS simulation revolves around three aspects of the simulation: the jobs, resources, and time. There are two categories of output data: Scheduling and Performance. Scheduling data quantifies the impact of rework and resource contention on job completion time. In order to visualize scheduling data, a chart needs to quantify the amount of time that a production process lasts in a scenario. An ideal chart for visualizing production processes is a Gantt chart. Each scenario has three Gantt charts displays: a job Gantt chart, resource Gantt

Figure 4. Data Flow Diagram

chart, and a Scenario Gantt chart. Performance data quantifies the impact of a particular task on job completion time. In order to visualize performance data, statistical estimators are computed by analyzing scenario data gathered over multiple scenario runs.

Scheduling Display

The Scheduling Display visualizes all job related data from a single scenario run on multiple Gantt charts. The display illustrates the effects of job arrival order and their subsequent interactions on job completion time. Each task within a job is displayed in two parts: a non-value added time (time waiting) and a value added time (time processing). Figure 5 shows an abstraction of display characteristics that appear on the scenario Schedule Display. The x-axis displays a time line of jobs in the scenario. Each job is distinctly color-coded, which can be customized according to the user's preference. The y-axis displays all of the jobs in the scenario.

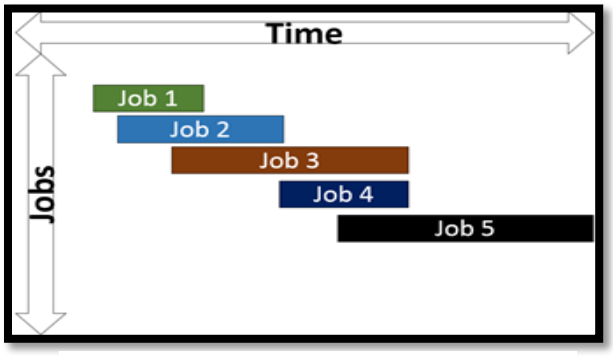


Figure 5. Scenario Gantt Chart

Specific job details can be seen from tips which will display from the user hovering over a particular job. However, for greater detail, the user can select a job, which displays a Gantt chart of the tasks and sub-jobs of the selected job, as seen in Figure 6 and Figure 7. At this level, the user can see the job's complete task flow. All tasks follow a two part color scheme for showing non-value added and value added time. These schemes can be customized but as an example, Figures 6 and 7 feature red for non-value added time and green for value added time. The Schedule Display facilitates the analysis of a scenario's performance.

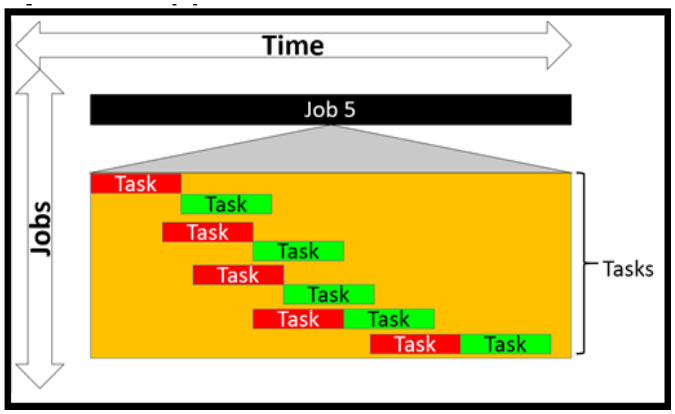


Figure 6. Job Gantt Chart

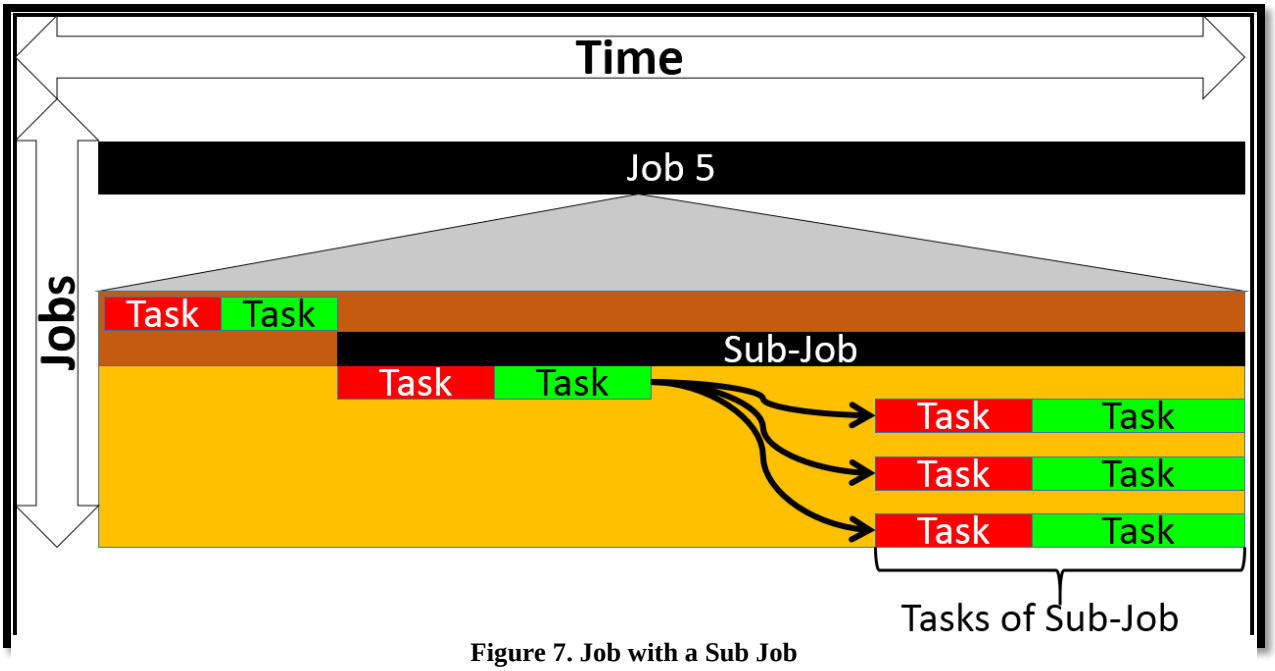


Figure 7. Job with a Sub Job

The resource Schedule Display displays the resource contention in the scenario. Figure 8 shows an example of a resource Schedule Display. The x-axis displays the time line of resource utilization in the scenario. The y-axis displays the resources, which are separated into individual sections. Each task will have a job name displayed immediately to the upper right of the task. Tasks will be distinctly color-coded according to its corresponding job. When a user hovers over a task, characteristics are shown in a tip box. The resource Schedule Display helps visualize the resource contention in a scenario.

Figure 8. Machine Schedule Resource Gantt Chart

Performance Display

A single scenario run within PPS is performed to understand the effects of choosing a specific schedule. A single run however, does not have any statistical significance and does not show the effects of probabilistic variables, such as rework and resource contention. To quantify these effects, the Multiple Run Analyzer is used to analyze a scenario that is simulated multiple times. Once these runs are completed, data relating to jobs, tasks, and resources like job completion time, resource processing times, and queue related data are computed. Analysis of these variables by the Multiple Run Analyzer allows for the computation of a confidence interval. Additionally, a minimum, maximum, and

average is appended to each appropriate variable to allow the user to gauge the limits of each statistic. The analyzer then formats the data and sends it to the Performance Display for viewing. The data is visualized in a textual table as seen in Figure 9.

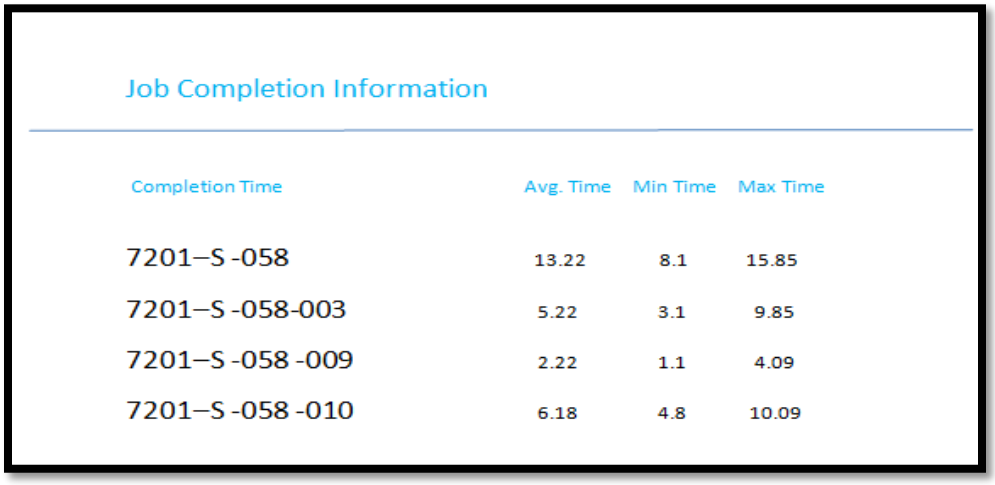


Figure 9. Performance Output Table

CONCLUSION

The PPS system is currently being produced as a prototype for NNI. The prototype will enable the user to better understand job and resource interactions and their effects on job completion time. The system architecture has been designed to not restrict the application to just the NNI-OP facility. Minor alteration of core PPS structures allows the PPS system to model other production facilities beyond the one discussed in this paper. Additionally, when new production machines are added to the facility, the PPS system can easily incorporate the new machines. Interaction between the SCDT and NNI-OP is ongoing to finalize the characteristics of the PPS system. Formal documentation of the PPS system is being drafted for facilitating future expansion of the PPS system's capabilities. These expansions first focus on automating many of the input processes of the PPS system. This would shorten the amount of time needed for creating new PPS scenarios. The incorporation of a real-time facility data collection system could be used as an input to start the simulation from a non-empty and idle state. These expansions will help reduce the time spent initializing the PPS system.

ACKNOWLEDGEMENTS

The authors would like to thank Newport News Industrial (NNI) for their participation and the valued input provided by James Whitley, Brian Bangs, and Erin Schiller. The authors would also like to thank Rob Lisle, Chris West, Irin Hall, John Lillard, Erick Hagstrom and the others from Newport News Shipbuilding (NNS), Department K76, for their support throughout the development process. The authors are especially grateful to Rex Wallen. He served as the subject matter expert and as the main interface between the SCDT, NNI, and NNS. Finally, the authors would like to give a very special thanks to Dr. Roland Mielke and Dr. James Leathrum. Without their support, expertise, and instruction, this would not have been possible.

REFERENCES

[1] Kádár B, Pfeiffer A, Monostori L. Discrete Event Simulation for Supporting Production Planning and Scheduling Decisions in Digital Factories. In: 37th CIRP Int. Seminar on Manufacturing Systems; Digital Enterprises, Production Networks. Hungary, 2004; 444-448.

[2] Ullman, J. D. (1975) NP-Complete Scheduling Problems, Journal of Computer System Sciences, 10, 384-393.