Proceedings of the 1994 Winter Simulation Conference
ed. J. D. Tew, S. Manivannan, D. A. Sadowski, and A. F. Seila

MODELING AIRCRAFT ASSEMBLY OPERATIONS

Harold A. Scott

Boeing Computer Services
Research and Technology
Seattle, Washington 98124, U.S.A.

ABSTRACT

A simulation model can be a powerful tool for understanding the complex interactions of aircraft assembly operations. An accurate model helps identify the effects of resource constraints on dynamic process capacity and cycle time. To analyze these effects, the model must capture job and crew interactions at the control code level. This paper explores five aspects of developing simulation models to analyze crew operations on aircraft assembly lines:

- representing job precedence relationships
- simulating crew members with different skill and job proficiency levels
- reallocating crew members to assist ongoing jobs
- depicting shifts and overtime
- modeling spatial constraints and crew movements in the production area

1 INTRODUCTION

Assembly of a commercial jet airliner is a very large and complex process. A typical aircraft (Figure 1) is assembled from millions of parts and goes through thousands of assembly operations.

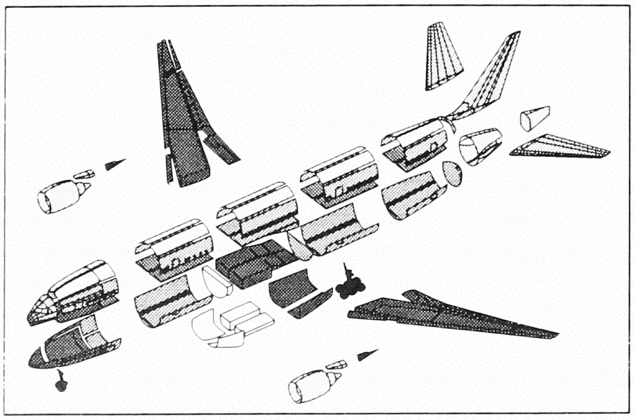

Figure 1: Aircraft Assembly

Thousands of suppliers from around the world provide components and subassemblies for final assembly. The success of this process depends on the effectiveness of thousands of skilled workers who assemble the aircraft. Discrete-event simulation models provide insights into dynamic relationships between crews and assembly operations to improve crew assignments and operations effectiveness.

This paper presents modeling and operational issues in simulating dynamic interactions of crews with jobs on an aircraft assembly line.

We will:

- present frameworks to represent aircraft assembly crew movements and job precedence relationships in a simulation model
- discuss modeling the allocation of crew resources
- introduce additional operational characteristics required to create a robust simulation model of the aircraft assembly process

2 AIRCRAFT ASSEMBLY PROCESS OVERVIEW

An overview of the aircraft assembly process will highlight some unique modeling issues. This paper focuses on crew operations in aircraft assembly and installation processes, comprising:

- airframe assemblies (such as wings, fuselage sections, empennage, and nose sections)
- airframe join (joining of the major assemblies)
- final assembly and installation (including interiors, landing gear, engines, systems, functional tests)

Figure 2 shows the basic components of an aircraft assembly line. Each aircraft is assembled in a sequence of control codes. A control code's cycle time can range from one to seven days depending on the line's production rate. At the end of each cycle, the assemblies and subassemblies are moved to their next control code. Major airframe assemblies are usually joined at a single control code. At this control code, the plane takes on its familiar shape.

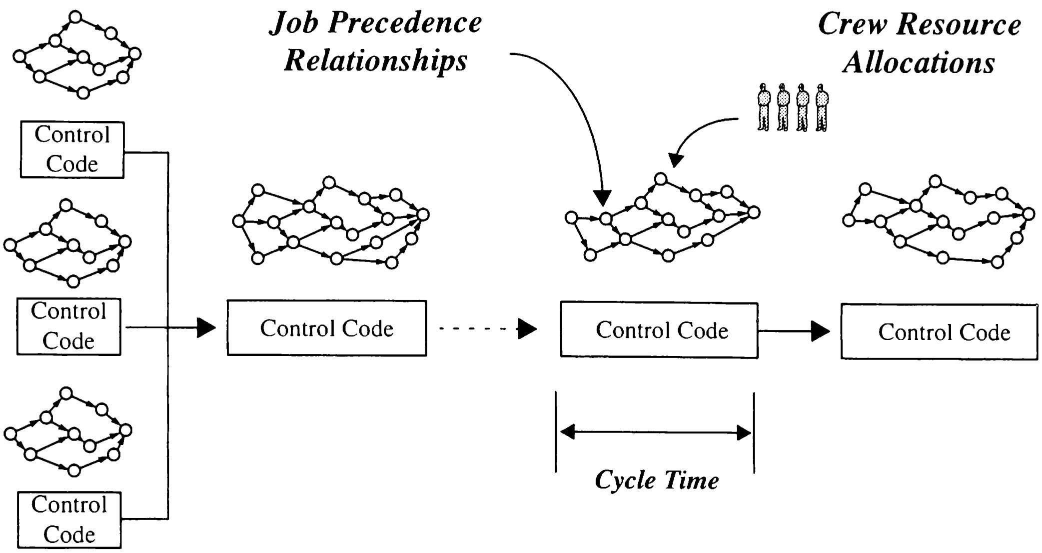

Figure 2: Aircraft Assembly Process

Within each control code, assembly crews complete up to several hundred jobs during each cycle. Most of the jobs have precedence relationships (such as the need to install seat rails before the seats). The amount of work and relationships between jobs can differ from cycle to cycle (for example, a passenger airliner might be followed by a freighter). Each job is located at a specific area around or on the assembly and/or assembly fixture. Crews go to these locations to perform the job. Cranes and crews also move material and tools to job locations.

Each job requires specific skill type(s). Job durations typically range from minutes to hours. Job duration depends on the number and proficiency of crew members assigned to the job. A minimum, normal or maximum number of crew members can be assigned to each job. Crew members finishing a job early can assist other crew members on other jobs or tasks (such as cleaning the area). Inspectors check once the job is completed.

3 REPRESENTING JOB AND CREW IN A SIMULATION MODEL

With most process-oriented simulation languages, the entire aircraft assembly line can be modeled as a series of control codes. In this representation of the assembly process, entities represent assemblies. Assemblies are routed through a network of queue/server nodes representing control codes. The assembly line rate determines an entity's duration time at each queue/server. To represent the joining of assemblies at a control code, the simulation logic combines entities into a single entity.

To evaluate crew effects on the dynamic capacity of the line, this model must simulate the assignment of crews to many available jobs at different locations around the aircraft assembly and tool. Each crew member's performance can be affected by:

- job precedence relationships
- work area space limitations
- total movement around the assembly area

This section presents a framework to model job precedence relationships, crew movements, and crew spatial constraints in the aircraft assembly process.

3.1 Job Precedence Relationships

Job precedence relationships within each control code (Figure 2) are the foundation of the aircraft assembly model. Representing the assembly and installation sequences, precedence relationships have an impact on crew resource allocation.

To model job precedences with a process-oriented simulation language, we first define job precedence relationships for each control code as an activity-on-node (AON) network. Next, we can represent each job on the AON network with a server module construct. An entity routed between two server modules symbolizes a precedence. An entity's arrival at a server module signifies completion of a preceding job. The server module holds arriving entities until preceding jobs are completed. The server module node will also branch entities to all dependent nodes.

At the beginning of a control code's cycle, we create one entity for each dependent node (job), then route each entity to a dependent node. An entity's arrival indicates completion of a preceding job. If the node has more than one precedence node, we hold the entity in a queue until all preceding job entities arrive at the node. When all preceding job entities have arrived, crew allocations determine the job duration. Upon completion of the job, we route the entity or entities to the next dependent node(s). We continue this process until the final entity arrives at the final node (i.e., the node without dependent nodes), indicating completion of a cycle.

A job precedence model can be built with the graphical constructs of a process-oriented language. However, when modeling a series of control codes, changes to the precedence network become unwieldy to manage. Project management software packages offer more robust methods for building precedence relationships for control codes. These packages also check for cycles in the precedence networks. (These cycles can be difficult to identify during a simulation run.) An event-oriented simulation model can address the indexing logic required to take advantage of the project management software's precedence relationship data structure. This is an important capability, since jobs and precedence relationships can change from aircraft to aircraft.

3.2 Crew Movements

Each job is associated with a specific position in or around the aircraft assembly area. In contrast to workers on traditional automobile assembly lines, crew members can cover a wide area in a day. The simulation model needs to address the movement of crews around the aircraft assembly. Distances traveled to and from jobs can have a significant impact on the capacity of the control code and assembly line, especially if the crew can be sent to jobs in other control codes.

We can take advantage of the transporter constructs found in most manufacturing simulation languages to model crew member movements. As a job becomes available on the precedence network, the job module requests a transporter type (crew member with a specific skill required to perform the job). Movement of a transporter entity from its current location represents a crew member walking to the job. Note the transporter can also be moved to another location en route to the requesting job's location. This action could represent the crew member picking up tools and/or materials for the job. Upon the transporter's arrival at the job location, the job begins and the transporter is kept in a busy state for the duration of the job. The transporter is free upon completion of the job.

The modeler must predefine the path distances and job locations for the transporters. Figure 3 illustrates example transporter networks around the aircraft with stations representing job locations. One network represents paths to floor level jobs, while the other represents paths to interior installation jobs in the fuselage. Nodes on the network path away from the aircraft represent material stores, tool rooms, and break areas. The networks also connect with other control code networks, as indicated by arrows.

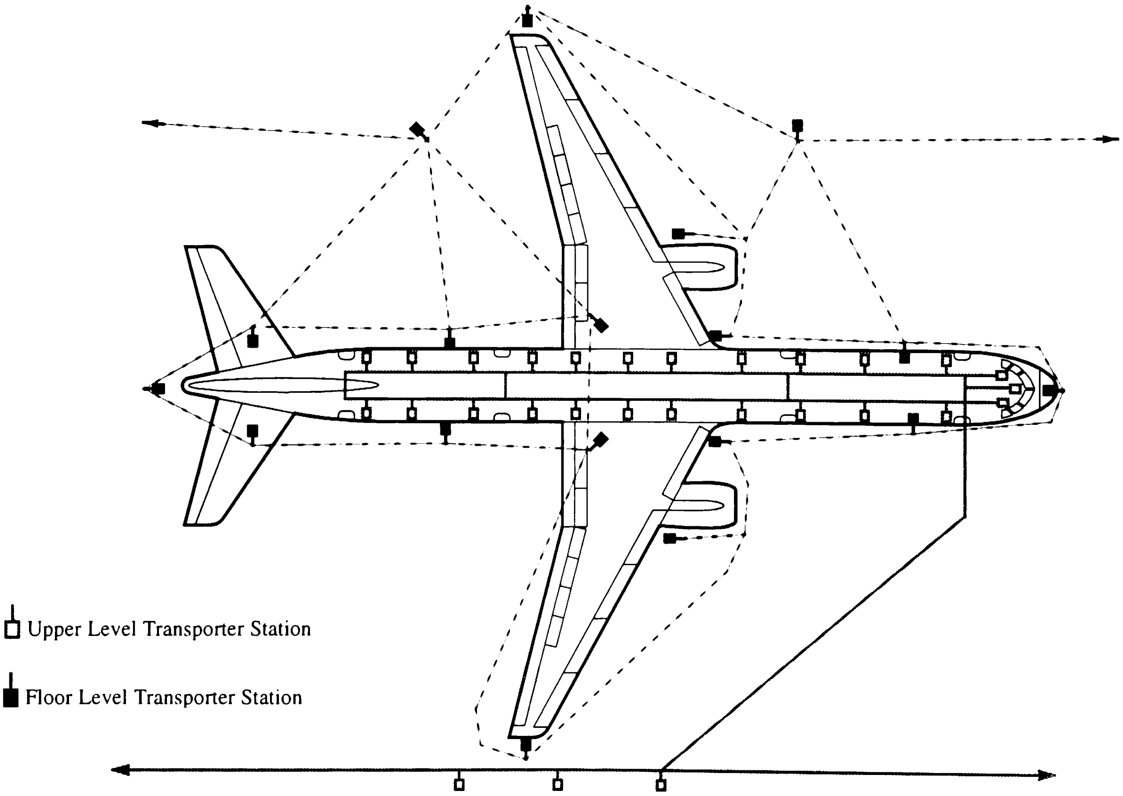

Figure 3: Aircraft Assembly Process

Crew transporters can help analyze crew movements for process improvement efforts. Distance and movement information from transporters' paths and actions can be used to determine wasted movement by crews. For example, a simulation analysis might show that a crew transporter makes a large number of trips to a tool room. Based on this analysis, small tools could be staged at the line to reduce the number of trips to the tool room, resulting in increased productivity. Distance information can also be used to analyze control code layout and material staging.

3.3 Spatial Constraints

While many jobs can be performed simultaneously in the same area of the aircraft, its physical size can limit the number of crews who can work in proximity (in the cockpit, for example). Thus, spatial constraints can affect a control code's capacity by limiting the number of crews assigned to a job. Consider an example where a maximum of three crew members can work in the same area, and two crew members are already working there. If two additional crew members are available for a job, only one can be assigned without violating the spatial constraint. This assignment limitation increases the job duration.

Using transporters to represent crews, we can simulate these constraints in the model by limiting the number of transporter stations in an area. For example, the cockpit area in Figure 3 has only three transporter stations. If all three stations are filled, another available transporter (crew member) cannot be sent to a requesting job in the area. The crew allocation logic must determine if the available transporter should wait for the job or be assigned to another available job.

4 CREW RESOURCE ALLOCATIONS

The simulation model needs to allocate the appropriate type and amount of resources (crews) to the job before processing. Crew resources can be allocated to requesting jobs either by a predefined schedule or based on the state of the system. This section will present several unique operational characteristics that make crew resource allocation a challenging task for simulating aircraft processes.

4.1 Allocate a Range of Resources to a Job

Each job requires specific skill type(s). A job can have a minimum, normal or maximum number of crew members assigned. Adding more crew members to the job will reduce the job duration at a linear rate until the number of crew members reaches normal (Figure 4). Beyond the normal level, job duration decreases at a much lower rate for each crew member added to the job.

Allocating a range of crew members can be problematic. Assignments can be determined by a heuristic (minimum critical ratio or Brook's algorithm, for example).

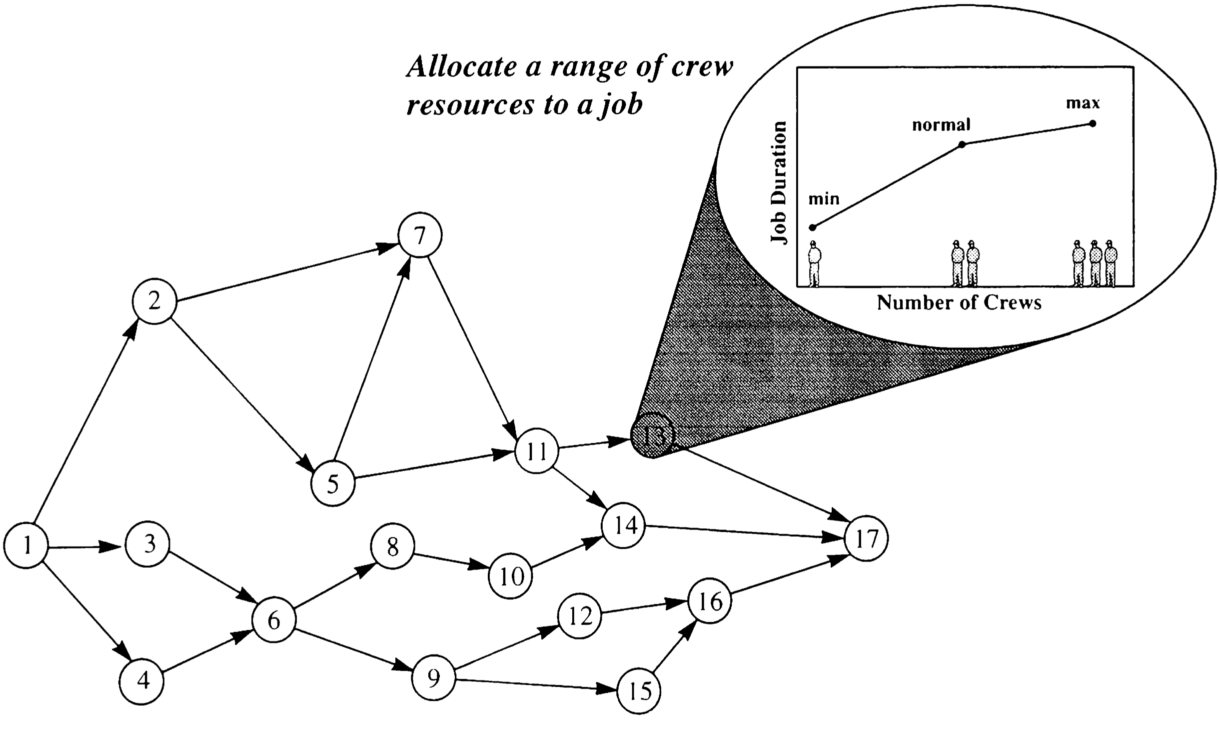

Figure 4: Allocate a Range of Resources to a Job

The heuristic must evaluate the processing time for multiple resources before choosing an assignment. Space requirements must also be checked. This takes considerable computer overhead during the simulation run.

Special care needs to be given to control event processing in the precedence network model when selecting the number of crew members to allocate to a job. Process-oriented languages continue to process an entity until a wait or hold state is reached. This event processing can have undesirable effects on crew allocation. For example, two jobs can start simultaneously after a preceding job finishes. The entity that is processed first might seize a maximum amount of resources based on simulation event processing logic. This crew assignment might not leave any resources to the second entity, which could be on the critical path. Extreme care must be given to processing simultaneous events on the simulation calendar when a range of resources can be assigned to a job.

Using a predefined schedule can reduce the crew assignment complexity. With crew assignments already determined, the simulation model just needs to check spatial constraints at the requesting job's area and resource availability at the scheduled time. If these two conditions are satisfied, the crew transporter moves to the job location and remains busy for the duration of the job, based on the crew member's proficiency level. This is an excellent method to evaluate a schedule for a control code.

4.2 Reassign Resources to Ongoing Activities

Aircraft operations enable the reassignment of crew members to assist ongoing jobs. This is an important activity to capture in the model because job durations range from minutes to hours. Crew assistance can affect capacity of the control code. Most manufacturing simulation languages cannot assign another resource to an ongoing activity, then change the processing time to reflect the additional resources.

Figure 5 is an example of the reassignment of a crew member to an ongoing task. The crew member completing job 4 at T_{now} cannot be assigned to any new jobs due to the job precedence constraints. Job 2 is an ongoing activity on the current critical path. A minimum of one and a normal of two crew members can be assigned to job 2. One crew member is currently assigned to the task and is scheduled to complete the task at simulation time T_2. Based on these conditions, the crew allocation rules assign the crew member completing job 4 at T_{now} to job 2.

To reassign the crew member completing job 4 to ongoing job 2, move the transporter representing the crew member to an available station for job 2. When the transporter arrives at the job 2 station, remove the completion event for job 3 from the event calendar. Calculate a new completion time for job 2 by first determining the amount of time remaining on the task with the current crew member. Then, extrapolate the remaining time to complete the job with two crew members. Compute a new completion time for job 2 with the two crew members. Add the completion time T_3 to the event calendar. Each transporter remains busy while assigned to a job and must be released when the job is completed.

Modeling crew reassignment requires extensive activity tracking and event calendar manipulations. Representing crew members as transporters instead of resources eliminates the reassignment of resources to the ongoing event when the event is removed and replaced on the event calendar. Usually, the simulation calls a routine written in a procedural language to execute the resource reassignment logic. Eliminating the tracking and reassignment of crew resources reduces the overhead in modeling this reassignment event.

4.3 Assign Crews Based on Shift Changes

Failure to include end-of-shift events in a simulation can affect the model's accuracy to predict a control code's capacity and cycle time. In most manufacturing simulation languages, resources must complete the job before being released at the end of a shift. This logic does not capture the shift change events that occur in the aircraft production process. Representation of crew members as transporters provides the flexibility to model shift changes.

The simulation logic uses a transporter's availability to represent a crew member's shift status. To represent a crew member ending a shift, the transporter's status changes from active to inactive. The simulation logic will not allocate an inactive transporter to requesting jobs. A transporter's status changes from inactive to active to represent a crew member beginning a shift. Before changing the status of a transporter, the simulation logic checks the status of each transporter to determine if the transporter is busy or idle. If the transporter is idle, the simulation logic changes the transporter's status to inactive, then schedules an event to change its status to active at the next shift.

If a transporter is busy, the job associated with the transporter must be reassigned to a transporter whose status has just changed to active. This event simulates the reallocation of a partially completed job from the previous shift to a crew member on the new shift. To reassign partially completed jobs to crew members on the new shift, we employ the same logic to assign crew transporters to ongoing job activities (discussed in the previous section). For each busy crew member completing a shift, the job completion event is removed from the event calendar. Crew members from the new shift are allocated to the job. The simulation logic calculates the new completion time

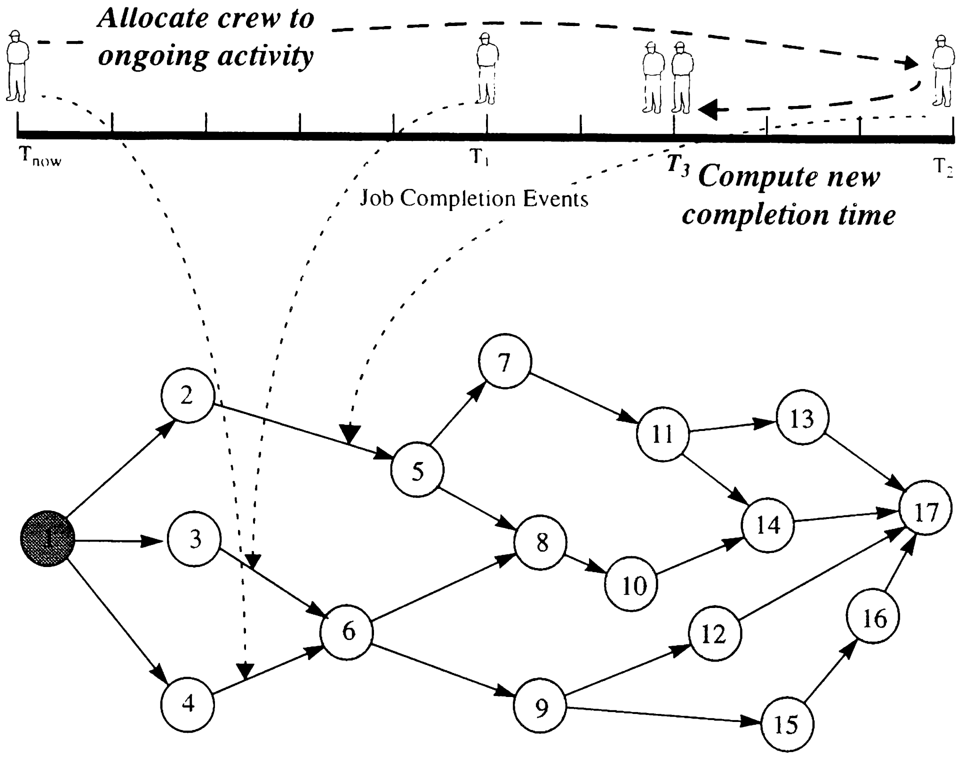

Figure 5: Allocation of Crews to Ongoing Jobs

based on remaining job time and the number of crew members allocated to the job from the new shift, then reschedules the event on the simulation calendar. Note this assignment logic can address differences in the number of crews and skill types between shifts.

We can also modify the shift change logic to eliminate partially completed jobs between shifts. This strategy significantly reduces overhead in crew shift change allocation logic, but also can significantly reduce the line's capacity. Using this approach, the simulation logic calculates the remaining time until the end of a shift at each job event. If the duration times of all available jobs exceed the time until the end of the shift, crew resources are assigned to housecleaning tasks. Otherwise, a crew member can be assigned to the next available job or assist an existing job. To calculate job duration, the model must account for the transporter's travel time to the available job's location.

The model's shift change logic must consider overtime. To model overtime, we can extend the transporter's availability over the normal shift time and incorporate the overtime crew allocation logic in the model. In an overtime situation, crew members on the next shift usually have to wait for the job to be completed by crews on the previous shift. When job precedences constrain the number of jobs available to crew members on the new shift, we can take advantage of the crew allocation method for assigning crews to ongoing tasks. Crew members on the new shift can be assigned to assist crew members on jobs from the previous shift. Using the transporter representation of crews, we also can analyze the space impacts when crews from two shifts overlap.

5 ADDITIONAL OPERATIONAL CONSIDERATIONS

In addition to crew representation and allocation methods, a model needs to capture other special characteristics of aircraft assembly operations. This section will briefly discuss key characteristics that affect assembly crew operations at all jobs. Many jobs have special requirements (such as chemical disposal in paint and seal areas), which need to be addressed on a case-by-case basis.

5.1 Learning Curves

Learning curves have significant influence on the time a crew member requires to complete a job. Unlike an automobile assembly line where an operator repeats a job in less than a couple of minutes, an aircraft assembly crew member may not return to the same task for days. Therefore, learning curves have a much greater impact on job duration and the line's throughput time. During a new model or process introduction, these curves are especially critical for predicting the line's cycle time at various points in time. If available, the modeler can use empirical data to define learning curves. Otherwise, a learning curve algorithm can be employed in the model.

5.2 Crew Proficiency Level

Besides skill type, each crew member has a proficiency level. The proficiency level of a crew member (identified as a skilled mechanic or journeyman, for example) also determines the length of time to complete a job. Including proficiency levels in the model adds another level of complexity to the crew allocation logic. The modeler might consider this factor when crew proficiency levels vary greatly or when evaluating schedules with the simulation model. Variations in crew proficiency level mixes can also be reflected in the distributions that represent job processing times.

5.3 Inspection and Rework

When a job is completed, crew members call inspectors to check their finished work. Once the work passes inspection, the job is considered complete. Job completion delays can occur waiting for inspectors. Note also that there can be many types of inspections and certifications including those by government agencies and airline representatives. Simulation model analysis can be used to determine the number of certified inspectors required for a given line's cycle time. If rejection rates are significant, a modeler also needs to include:

- failure rates by job type
- rework time
- material disposition delays

5.4 Resource Constraints

Availability of tools and cranes can affect the capacity and flowtime of the aircraft assembly line. If a crane or tool is not available at the beginning of a job, the part cannot be moved to the job location. If these resources are found to be potential bottlenecks in the process, they need to be captured in the model. The method to model these resources needs to be evaluated based on resource type and activities.

5.5 Additional Crew Activities

Meetings and training classes can significantly reduce the number of available crew members for assignment to production jobs. A robust model of a control code also needs to allow for lunch and break activities. Vacation can be included in the model and its impact evaluated for peak vacation season. Sick leave projections can be based on empirical data and modeled as random events. These factors affect the capacity of the assembly process. Planners can change these factors to minimize effects from these activities.

6 CONCLUSIONS

Discrete-event simulation models can provide insight into the dynamic capacity and performance of an aircraft assembly line resulting from interactions between resources and jobs. A simulation analysis can facilitate understanding effects of these interactions on a line's throughput time.

Using a simulation model of their areas, crews can evaluate and discuss changes in schedules and assembly sequences with planners before implementation. By incorporating learning curves into the model, planners evaluate the effects of new model configurations, changing the aircraft production sequence, and rates on performance and costs. Most important, crew members can employ simulation models to test and evaluate their process improvement ideas before implementation on the line.

This paper presented methods to represent and model aircraft assembly job precedence relationships and crew operations. Using transporter constructs to represent crew members' activities is a convenient way to model unique spatial constraints of the aircraft assembly line. This representation also enables modeling crew allocations in the aircraft production process. Most simulation packages can illustrate transporter movements with animations, allowing visual evaluation of crew movements and space constraints.

In trying to capture all the significant states of the system, the simulation model can grow in size, complexity, and execution time. With the crew transporter representation, most models will create an entity for each transporter and the active and completed jobs in the precedence network. Just one control code with hundreds of jobs and crew members creates a very large model. Frequently the most cost-effective approach is to identify and model the perceived bottleneck control code first before modeling an entire line.

A modeler faces many challenges in modeling and analyzing aircraft assembly crew operations. As with all modeling projects, a major portion of the effort is spent collecting data and understanding the process. A robust model of the process can be well worth the effort in evaluating the dynamic capacity and performance of aircraft assembly operations.

REFERENCES

Badiru, Adedeji B. 1993. *Quantitative Models for Project Planning, Scheduling, and Control.* Westport, Connecticut: Quorum Books.

Baudin, Michel. 1990. *Manufacturing Systems Analysis with Application to Production Scheduling.* Englewood Cliffs, New Jersey: Yourdon Press.

Law, Averill M. and Kelton, W. David. 1993. *Simulation Modeling & Analysis.* 2nd ed. New York: McGraw-Hill, Inc.

Nahmias, Steven. 1993. *Production and Operations Analysis.* 2nd ed. Homewood, Illinois: Irwin.

Niu, Michael C. Y. 1988. *Airframe Structural Design.* Hong Kong: Conmilit Press Ltd.

Watkins, Kevin. 1993. *Discrete Event Simulation in C.* London: McGraw-Hill Book Company.

AUTHOR BIOGRAPHY

HAROLD SCOTT is an associate technical fellow of The Boeing Company. He received his Ph.D. in Industrial Engineering and Operations Research from Virginia Polytechnic Institute and State University. His undergraduate degree is from Georgia Institute of Technology. His main research interest is developing decision support technologies to improve the quality of manufacturing processes.