Proceedings of the 1991 Winter Simulation Conference
Barry L. Nelson, W. David Kelton, Gordon M. Clark (eds.)

AN APPROACH TO THE CONSTRUCTION AND USAGE OF SIMULATION MODELING IN THE SHIPBUILDING INDUSTRY

Mark K. Brazier

Systems Modeling Corporation
The Park Building
504 Beaver Street
Sewickley, PA 15143

Carl J. Perry

Department Of Industrial Engineering
Ingalls Shipbuilding
Post Office Box 149
Pascagoula, MS 39568-0149

ABSTRACT

This paper describes a two-level modeling approach for developing simulation models in the shipbuilding industry. At the shop floor level, a series of "low-level" models simulate the behaviors of individual hull components as they are fabricated and processed through the shop. Output from one model is used as input to the next in accordance with the appropriate manufacturing sequence. A single "high-level" model simulates the overall shipbuilding process, modeling the manufacturing of major assemblies as they are fabricated. Both levels are schedule-driven to allow for the analysis of a proposed schedule with respect to capacity requirements, inventory, throughput, etc. In addition, each is animated to depict graphically their behavior.

A parallel development effort of the high-level model with the low-level models has provided both an initial rough cut analysis tool for forecasting and planning as well as a framework for integration of the low-level models into a single implementation for detailed macro analysis.

Included with this paper is a description of the Ingalls manufacturing environment as well as a discussion of the management issues involved in the development of a resident simulation team.

1 INTRODUCTION

Starting in 1938, Ingalls Shipbuilding has been one of the most modern and competitive shipyards in the country. Competing worldwide for limited new vessel orders, Ingalls Shipbuilding has grown to over 800 acres in size, and now employs approximately 16,000 people--in a business that has realized over a 70 percent decline in its domestic industrial base during the past decade alone [Bergstrom, 1991].

Essential to the success of building a product with the size and complexity of a small city is the ability to plan and execute near-term work in great detail while also planning for future work at a level of abstraction sufficient for operational planning. A typical ship can take three to five years to complete, requiring the efforts of thousands of people during its construction. This planning process is further exasperated by the dissimilar nature of the product mix. Ingalls currently has four different ship classes under construction within the same shipyard. The Wasp class multipurpose amphibious assault ship is over 840 feet in length and weighs in excess of 40,000 tons. The Ticonderoga class is 567 feet in length and displaces nearly 9,500 tons. Finally, the Arleigh Burke and SA'AR 5 Corvette classes follow at 504 feet, 8,300 tons and 281 feet, 1,200 tons respectively.

Because Ingalls utilizes a technique known as inverted modular construction in the building of its ships, the allocation and use of plant space, manpower, and material handling equipment must be orchestrated with great precision, both in daily planning and in the development of new contract proposals. To assist in these planning processes, several inventory control and operation research technologies have been incorporated by the manufacturing management staff throughout the years. Now, simulation will give Ingalls another tool to manage facilities with a far greater level of resolution than ever before. This increased capability will answer questions regarding capacity allocation and scheduling down to the shop floor level, significantly increasing Ingalls' competitive posture.

This paper describes the development of dual-level computer simulation models designed to accomplish precisely these objectives. The high-level model is designed to simulate the production planning and control operations for the entire shipyard at the macroscopic level. Whereas the high-level model is a low resolution representation of the entire shipyard, the low-level models are designed to simulate each of the individual work centers at a much more significant level of detail. These models are designed to analyze specific machine and manpower behaviors, effects of schedule changes on shop loading, equipment failure, etc.

2 PROJECT MANAGEMENT

Discrete event simulation is a concept that is fairly new to the shipbuilding industry. As with any new concept, there is always the question as to whether the expenditure of funds and manpower will yield the desired results. Because the project at Ingalls required a product with modeling flexibility, animation capabilities, and the ability to utilize information from the company's database, SIMAN/CINEMA were selected as the language of choice.

While the choice of a simulation language was important, perhaps a more difficult management choice was the selection and development of a resident simulation team. The first step in this process was identification of project skill requirements. Members had to be able to communicate to both management and production personnel, have an understanding of the production engineering and business planning processes, as well as the company mainframe computer system and the production planning and scheduling systems it hosted. They also had to understand how to gather data from the shop floor, run statistical analysis, and document it in a format that was both understandable and usable. Finally, they had to be able to learn and to implement a simulation language.

To meet all of these criteria, a team of personnel from both Production Control and Industrial Engineering was chosen. The team size was initially limited to six members in order to provide as much cohesiveness and group dynamics as possible. Members were asked to cross train each other in their various disciplines during the course of their work to provide as much understanding of each other's area of expertise as possible. In addition, each was asked to keep the others abreast of his or her progress throughout the model development phase.

In order to kickoff the project with as much momentum as possible, a decision was made to employ the services of the consulting group of Systems Modeling. An intensive two-week training course in the SIMAN simulation language was provided on-site. This training, coupled with additional project start up assistance discussed below, provided a solid springboard for the development of a simulation center for technical excellence within Ingalls.

Upon completion of the requisite simulation language training, the team was tasked to develop a functional specification defining the specific design structure and features that the proposed simulation model(s) would incorporate. This document served three purposes. First, it quantitatively defined management's objectives for the model at the inception of the project. Second, it provided a consolidated functional description of the environment that would be modeled. This description included supporting data defining machine process times, failure rates, material handling travel speeds and distances, queue capacities, etc. And third, it stipulated the specific design structure and features, data inputs and outputs, and assumptions used in the construction of the model.

The first step in development of this specification was obtaining management's project goals. Stated simply, the simulation staff was to develop a model or series of models that: (1) could be used to evaluate work flow and capacity utilization performance for a proposed schedule in each of the five primary production areas, and (2) could be used as a capacity forecasting tool in new business development.

Two underlying considerations became apparent to the modeling team at this time. First, the competitive nature of the shipbuilding market generated an immediate need for a forecasting tool. Second, senior management's acceptance and continued support of simulation could be enhanced measurably if a model or models meeting their needs could be demonstrated and delivered sooner than anticipated.

A requirement existed then for a model that could be used both for detailed analysis of the individual manufacturing processes and for macroscopic examination of proposed production schedules. These requirements, coupled with the desire to produce a meaningful product in a short time frame, resulted in consideration of a dual-level modeling approach. The detailed, or "low-level" approach, would provide a model or models that simulated the behavior of individual work centers at the required shop floor level of resolution. This approach, due to its higher precision and additional data requirements, would be more long term in nature. The second level, or "high-level" approach, would provide an immediate overall model of the shipyard, but initially at least, at a fairly simple level that approximated the shipyard processes. This model would be used as the initial forecasting tool.

Concerned about the magnitude and risk associated with pursuing a single low-level model approach, Ingalls management explored the possibility of developing the model in smaller and more manageable sizes. According to Pegden et al [1990], there were basically two ways to keep the simulation model "small." The first way was through "simplification and reduction." The second way was through "partitioning" or division of the proposed environment into smaller sections or modules. While the initial high-level model could clearly utilize the first approach, it appeared that the objectives of the low-level model would be best met by the second strategy, perhaps in conjunction with the first.

Given the work center structure at Ingalls, serial partitioning of the low-level model by organizational lines seemed the most logical choice. Each work center had well-defined boundaries that were understood by both management and the simulation staff. Since material flow began in the fabrication shop, this work center was selected as the first low-level model.

3 THE INGALLS MANUFACTURING ENVIRONMENT

Traditionally, shipyards have been designed around a series of slips and ways. When a ship was built in these yards, the yard would "strike" the keel, build up the hull, and then outfit the inside--from the inside. While this method worked, it required that much of the outfitting of pipes, ventilation ducts, lighting fixtures, and so forth be accomplished in an overhead position that was often cramped, poorly ventilated, and poorly lighted. It made much more sense to build small sections, pre-outfit them while upside down or in a downhand manner, turn the sections upright, build sections into modules, and finally assemble modules into ships. This method of manufacturing is known as "inverted modular construction," and it is the choice of techniques at Ingalls Shipbuilding.

Built in the late 1960s, Ingalls West Bank Shipbuilding Facility is divided into five primary production areas. These areas are the: (1) production shops, (2) outfitting area, (3) erection area, (4) module integration area, and (5) wetdock area (see Figure 1). Each of these areas and their interrelationships is discussed below.

Production Shops--The seven major work centers that comprise the production shops area are the: (1) steel fabrication shop, (2) aluminum fabrication shop, (3) steel panel shop, (4) steel shell shop, (5) pipe shop, (6) sheet metal shop, and (7) the electrical shop. These shops are responsible for conversion of raw materials into units called assemblies.

The steel fabrication shop is considered the "pacing" shop in that it is the first shop to begin work. Materials manufactured from this shop can flow to any location within the shipyard, but it primarily feeds the panel and shell shops as well as the outfitting area. Output consist of subassemblies such as machinery foundations, shell plates, deck sections, bulkheads, masts, etc.

The panel shop is responsible for butt welding deck and bulkhead plates together to form larger sections. T-beams are then added for structural support and the completed subassembly sections are transported, as required, to either the shell shop or the outfitting area for assembly integration.

The shell shop is dependent upon both the panel shop and the fabrication shop for its material input. This shop is responsible for the integration of bulkheads, decks, and shell plates into assemblies. Completed assemblies are then transported to the outfitting area--if pre-outfitting is required--or to the erection area for module integration.

The pipe shop manufactures pipe assemblies for the cooling, steam, fresh water, fuel, and waste systems of the ship. These assemblies are delivered to either the outfitting or erection area for fitting into the appropriate hull assembly.

The sheet metal shop manufactures ventilation ducting for the ship. These components are also supplied to either the outfitting or erection area for fitting into the appropriate hull assembly.

The electrical shop is the final shop in the production shops group. It is responsible for the production of the power panels and small electrical equipment foundations not built in the fabrication shop. In addition, it prepares electrical components installed aboard the ships. All completed electrical components are transported either to the outfitting, erection, or module integration area for installation.

Outfitting Area--This area is responsible for pre-outfitting as much of an assembly as possible before its movement to the erection area for assembly integration with other assemblies. Pre-outfitting is done with inverted assemblies to provide a more natural downhand position. In this area, the pipe, ventilation, and electrical items located in the overheads of the finished ship will be installed. These pre-outfitted assemblies are then moved forward to the erection area.

Erection Area--Assemblies are delivered to this area in a vertical or "shipshape" orientation from either the outfitting area or the shell shop. This area is responsible for integration of the assemblies into modules--typically four to five per ship (Bays 1-5). Any remaining outfitting for deck located components, now downhand, is also completed in this area.

Module Integration Area--This area is responsible for integration (welding together) of the modules formed in the erection area. Pipe, electrical, and sheet metal crafts join the inter-modular systems while the hull craft joins the structure. At this point the ship is nearly

70 percent complete. It is then transported by electrical rail cars to the drydock where the vessel is lowered into the wetdock area.

Wetdock Area--The wetdock area is the final stop in the construction of the ship. In this area, crafts perform final installation and systems checks. Upon completion, the ship begins sea trials.

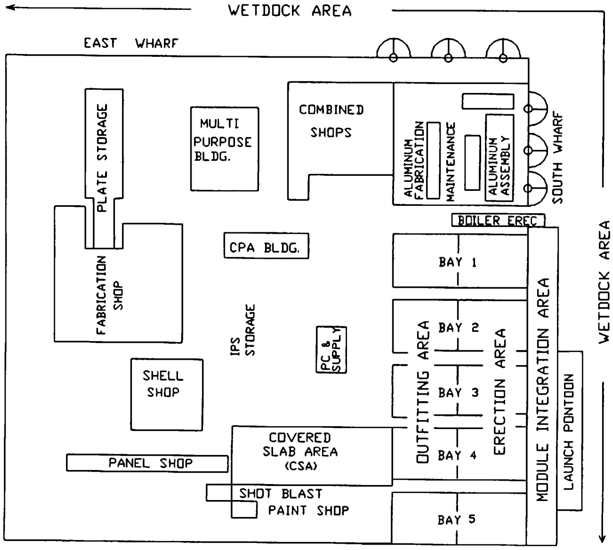

Figure 1. Ingalls Production Area--West Bank Facility

4 DATA ANALYSIS

At any given time, a work center may have hundreds of assembly work order packages or "bills" in process on the shop floor. These packages, generated weekly by computer and coordinated through production control, describe what assembly the work center is to produce, when it is to start production, when the assembly is due, and how the assembly is to be produced. In addition, these bills coordinate the assemblies routing throughout the work center. In the process of execution, each bill may direct the manufacture and assembly of hundreds of piece parts, each with its own routing sequence. To simplify the representation of this environment, data aggregation and simplification techniques were applied.

4.1 Data Aggregation

To reduce the number of assembly work order packages or bills that were explicitly represented within a given model, a FORTRAN program was developed to isolate bills into two categories called "noise" or "major." This program extracted historical data from the Ingalls database for assembly work order packages processed through a specified work center during the past four years.

Following data extraction, the program separated the packages into two groups based on man-hours required for completion and a user-defined "noise" level threshold value. For example, a threshold value of 900 man-hours would separate the data into: (1) "major" assembly work order packages requiring more than 900 man-hours, and (2) "noise" assembly work order packages requiring 900 man-hours or less. Using this method for different threshold values, the FORTRAN program calculated the number of bills that would fall into each category. A target of 400 "major" bills was established as the desired goal since it represented the maximum number of bills that could be reasonably evaluated in detail by the development team. Analysis of the fabrication shop indicated that a threshold value set between 300 and 500 man-hours would achieve this goal.

To insure that excessive aggregation would not occur at the higher 500 man-hour level during production periods of small man-hour assembly work order packages, a second FORTRAN program was employed. Using the Ingalls database described above, this program calculated the total number of man-hours expended by each bill and the calendar work-in-process span time required. Dividing man-hours by span time resulted in a weekly average for man-hours consumed. This averaging process was repeated for each assembly work order package, and a table was generated for the cumulative number of man-hours consumed by week by bill for the four-year period.

Repeating this process for the assembly work order packages identified in the "noise" category, populations and comparison plots for man-hours allocated to "noise" and man-hours allocated to all bills were made (see Figure 2a). Since a lower noise threshold value would place fewer bills into the noise category, the lower range limit of 300 man-hours was selected for comparison (see Figure 2b).

The noise threshold of 500 man-hours provided a measurable increase in aggregation based on the spread between the "all bills" and "noise" lines. This is especially noticeable during periods of productivity where assemblies are scheduled with smaller man-hour requirements (right-hand side of each plot). As can be seen, much of the production at this point would be included with the noise making explicit representation within the model difficult. Thus, for the fabrication shop, the lower threshold value of 300 would insure more explicit representation for a given schedule than would the 500 level, while still approaching the 400 bill target.

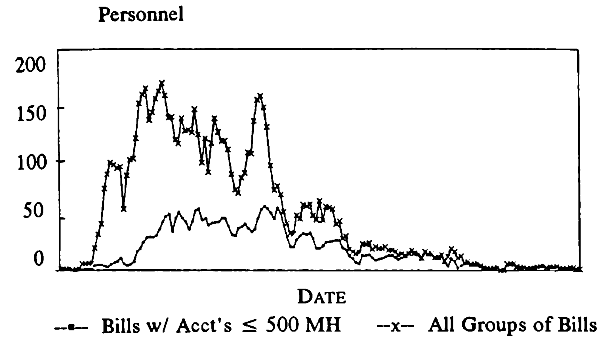

Figure 2a. 500 Man-Hour Fabrication Shop Manpower Profile

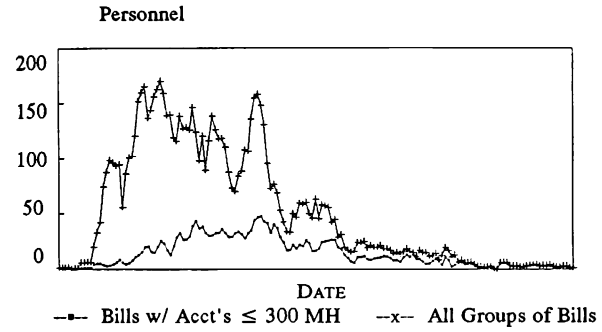

Figure 2b. 300 Man-Hour Fabrication Shop Manpower Profile

4.2 Data Simplification

Having separate material routing sequences for each work order assembly package would require a considerable amount of computer data storage space to maintain. As an alternative, consideration was given to the use of standardized route families, if they existed. Analysis of a selected group of assembly work order packages indicated that many of the manufacturing operations performed were identical in sequence. In some cases, packages that at first appeared unique in their routing, upon additional analysis, actually distilled into percentages of two or more previously identified routes.

Using a FORTRAN program developed to automate the route family identification process, a route sequence scheme was generated for each bill number identified. This data, organized by work center, was associated with its hull type and downloaded to diskette as a "hull assembly data file" for subsequent inclusion within the simulation model.

5 FORTRAN CODE

FORTRAN programs were utilized in two areas. To interface with the Ingalls mainframe computer, programs for production of schedule and analysis of data (previously discussed) were created. To interface with these files, additional workstation-based FORTRAN programs were developed for use by the SIMAN models.

5.1 Generation of SIMAN Schedules

Ship schedules generated by Ingalls are developed initially by moving backward in time from the desired delivery date and then establishing major event completion dates. These major events (in backward order) include: sea trials, main engine light-off, electronics light-off, drydock float-off, module integration, setting the house (superstructure), erection of major assemblies, pre-outfit of major assemblies, fabrication of major assemblies, and fabrication start. This process generates an initial schedule known as the major event schedule. In addition to this initial schedule, a detailed assembly erection schedule is developed to support the production dates listed in the major event schedule.

From these two initial schedules, production managers, using the PERT-based scheduling system, begin backward scheduling of assembly work order packages or bills. Ship drawings are evaluated by the planners, and each piece part of the ship is allocated to a specific bill. Each bill generated is assigned a schedule completion date based on its sequence relative to the master schedule. The start date of a bill is established based on its scheduled completion date and the estimated number of production days required for its completion. When completed, the scheduled bills form the master construction schedule that is maintained by the Ingalls mainframe computer as part of the production planning database.

Once a ship begins construction, weekly work schedules are produced for each production area based on the start dates for their respective assembly work order packages. These reports itemize bills that are scheduled for start work, list their due date, and list any bills that are in process or delinquent.

Because schedules used within the shipyard are not in a format directly transferrable to the simulation environment, a FORTRAN program was created to convert them. This program reads the appropriate schedule information directly from the production planning database used for master construction schedule generation. It then adjusts the original schedule from calendar time to equivalent simulation time based on a user-defined reference start and stop date.

Based on the user's inputs, the schedule conversion routine first removes holidays and partial work days from the master construction schedule beginning with the start date and ending with the stop date defined. Following these adjustments, the schedule is then compressed into simulation time using the specified start time as time zero. The modified schedule is then sorted by the adjusted assembly start day into an ascending chronological order with a secondary sort on due date. This completed schedule file is then downloaded to diskette for subsequent use by the simulation model.

The simulation schedule file is represented as a flat ASCII file containing five columns of data. These five columns consist of the hull number, bill number (also called assembly work order packages), start date, due date, and standard work hours for the assembly being introduced. In order to differentiate between different work center bill numbers, a unique work center identification number has also been included with each of the bill numbers. This identification number allows the same schedule file to be run on each of the low-level models as well as the high-level model without modification.

5.2 SIMAN Subroutines

The standard approach for building a SIMAN model requires that the specific variable values associated with the model be entered in the experiment frame. Any changes to these values require a recompilation of the experiment and a relinking before execution of the modified model can begin. However, due to the large volumes of data involved within the Ingalls manufacturing system, an alternative approach was developed to simplify the data entry process.

This approach allows the user to interface and customize the model before each run without actually changing the model itself. This is achieved through a configuration file that allows the user to define data at the beginning of each simulation run. A specialized FORTRAN subroutine named PRIME.FOR ("Prime") reads this file at the beginning of each simulation replicate. No recompilation or relinking of the model by the user is necessary. The user simply modifies the configuration file to represent different hull configurations, schedules, number of hours in a work day, noise threshold value, etc., and executes the model again.

Interface with the schedule during execution of the model is accomplished through a second FORTRAN subroutine named EVENT.FOR ("Event One"). Event One is initiated by a logical control entity created within the model at time zero. Once called, Event One reads off the next assembly work order package or bill record on the schedule file. Included with this record are the standard work hours for the bill. Based on the noise level threshold value input earlier by the Prime subroutine, Event One determines whether the bill is a "major" or a "noise" category assembly work order package. Standard work hour values greater than the noise threshold are categorized as major assemblies. These packages are explicitly represented within the model based on the data lookup and attribute assignment process described below. Those packages with time values less than the threshold value are represented as noise. These packages are inserted directly into the model following an "input window" entry process (also described below) for their descriptive attribute assignments. These assignments will be based on a distribution representing the aggregated noise work order packages identified during the data analysis process described earlier.

Major assembly work order package characteristics are defined within the hull assembly data file previously input by the Prime subroutine. Event One attempts to match the work center ID portion of these bill numbers to these files. For a low-level model, a match is necessary to determine the applicability of the bill number to that shop or model since the same shipyard-wide schedule is generated for all models. For the high-level model, a match is needed to determine which shop or work centers hull assembly data file is being used. Once a match has been found, a second search within the file itself is made for the assembly-specific data associated with that assembly work order package.

If a bill number match is found, an entity representing the major assembly work order package is created and the appropriate attribute assignments are made. These assignments include bill number, due date, major family route sequence code, standard work hours, and any additional information relevant to the needs of that model (e.g., the fabrication shop model requires attribute assignments for plate and shape counts). If a match is not found, an error message is generated describing the offending bill number, and the model is terminated.

Regardless of which category an assembly work order package falls into (major/noise), Event One must determine if the scheduled start time for the package is within the "input window" for the work center involved. The input window is computed as the current model time plus the look-ahead window defined by the user in the initialization file. If the scheduled start time is outside this input window, an additional attribute assignment equal to the scheduled start time minus the computed input window time is made. An entity with this assignment, upon model entry, is delayed until its input window begins. Once an entity has entered the model, or after it has completed its entry delay, it is assigned an arrival time and begins processing.

To prevent schedule reading during the input window suspension period described above, an equivalent delay for the read control entity is also incorporated. Thus, both entities will complete their delays simultaneously. Upon delay completion, the control entity again calls Event One, and the schedule input process continues. When the schedule has been completely read, the read control entity is disposed of and the simulation replicate terminates once the physical entities have cleared the system.

6 FIRST LOW-LEVEL MODEL--THE FABRICATION SHOP

The steel fabrication shop is literally the starting point for production of a ship. Responsible for the fabrication of all steel structures, its input consists of raw material that is cut, shaped and welded into the next most fundamental component--bulkheads, deckplates, foundations, support assemblies, piece parts, etc.

When an assembly work order package is schedule to start processing within the shop, raw material is loaded through the plate blast line (Bay 1) or the shape blast line (Bay 4). This material consists of two stock types called plates and shapes (flat bar and I-beams).

Regardless of which input line is used, all raw material is shot blasted to bare metal, primed, dried, and routed to its next operation as defined by its assembly work order package. As material moves through the fabrication shop, the work order is fragmented into various work-in-process groups throughout the shop's five bay areas and outside storage facilities. Upon completion of processing, the fragmented components are distributed, in a piece-meal-like fashion, to the various shops and downstream areas for additional processing and assembly.

6.1 Fabrication Shop Model Architecture

Based on the user-defined noise threshold value, work order assembly package entities enter the fabrication shop model in one of two methods. For major category packages, two entities are created for insertion into the model. These entities represent the plates and shapes portion of the package, each having a specific plate or shape count, bill number, due date, and route sequence code attribute assignment. These two entities, following any start delay for input window adjustment, are inserted into SIMAN model queues representing the receiving areas for Bay 1 (plates) and Bay 4 (shapes). Each entity will be explicitly represented in its routing sequence through the model.

For noise category packages, a single entity is inserted into a submodel portion of the main SIMAN model. Following any start delay for input window adjustment, an attribute assignment is made for the bill number. This assignment is based on a discrete probability distribution representing the aggregated family of assembly work order packages previously identified as noise during data analysis. Based on the "family" bill number and arrival time assigned, additional attribute assignments for due date, plate or shape count, and route sequence code are made. Following these assignments, the entity exits the submodel and begins processing at the first station in its route sequence. Because the original assembly work order package is represented as a member of an aggregated family, its real identity is lost. However, its use statistically for emulation of small bill impact on system performance is preserved through the Central Limit Theorem.

Although assembly work order packages are introduced into the shop in a chronological order based on their scheduled start date, processing priority within the shop is based on due date. To achieve this effect, queues within the fabrication shop are modeled using a SIMAN "Low Value First" (LVF) ranking discipline. Assembly work order packages arriving to these queues having the smallest due date attribute (due sooner) are thus released first.

To achieve the effects of individual plate or shape processing on capacity utilization while minimizing the number of entities within the system, an entity expansion and consolidation technique is employed. Since entities first entering the system represent an assembly work order package, a decomposition to individual subcomponents entity is necessary before the package can begin processing. To achieve this, the plate or shape assembly work order entity spawns children each time the resource for which it has queued becomes available. Each child is identical to its parent's characteristics except that its plate or shape count attribute is set to one. As these children are spawned, the parent's plate or shape count attribute is decremented appropriately. When this count reaches zero, the parent assembly work order entity is disposed.

Spawned entities flow through the model as individual plates or shapes until they arrive at the next queue in their process sequence. Upon queue entry, the entity checks for the presence of siblings with identical attribute characteristics. If the check is unsuccessful, the entity remains within the queue and assumes the roll of an assembly work order entity. If the check is successful, the plate or shape count of the older sibling is incremented by one and the arriving entity is disposed.

To represent the multiple route alternatives that an assembly work order package follows during its production sequence, a routing scheme is used. This scheme is based on the route sequence code assigned to the assembly work order entity when it was first created. Using this code as a SIMAN parameter set reference pointer, a route family can be associated with the subcomponents of the assembly work order package. For example, a code value of one would imply that the arriving assembly work order package follows the route family identified in parameter set one. Parameter set one would be represented as a series of percentages and SIMAN sequence set number pairs. These values, called routing pairs, represent the paths taken by the assembly work order package subcomponents during the production process.

Actual assignment of the SIMAN sequence set number occurs during the first assembly work order package decomposition. Using the SIMAN discrete probability distribution and the entity route sequence code attribute, a reference is made to the parameter set containing the route family data. Based on the cumulative probability percentages within this parameter set, a sequence set number is selected and assigned to the SIMAN "NS" attribute of the newly created child entity. Using this number and the "SEQ" option on model transfer blocks, SIMAN automatically tracks and updates progress of the entity through the model.

Because assembly work order package throughput statistics are required for analysis of schedule performance, a method is needed to provide the start and completion times of each assembly work order package processed. While SIMAN provides a special block construct for measurement of throughput statistics, the piece-meal-like departure of the assembly work order package subcomponents makes determination of final completion time difficult. To solve this problem, a logical entity is created and placed in a detached queue when the physical entity representing the assembly work order package begins processing. As the individual plate or shape subcomponent of the physical entity completes its processing, the plate or shape count of the logical entity representing it was decremented accordingly. When the logical entity plate or shape count reaches zero, the physical entity is considered to have finished processing and the logical entity is disposed. By measuring the life span of the logical entity, rather than the individual subcomponents, a throughput statistic for the original assembly work order is obtained.

As each individual plate or shape entity completes its processing, statistics regarding the subcomponents characteristics are saved. Using the SIMAN write block, attribute values and time of departure of the entity are written to a formatted, sequential file for post-model analysis and for input (inventory) to the next low-level model in series.

7 HIGH-LEVEL MODEL

The primary function of the initial high-level model is forecasting the effects of proposed schedules on manpower and work-in-process storage space utilization. Understanding what impact the scheduling of a contract will have on manpower, capital equipment, and work-in-process storage space is critical to efficient management of these resources. To provide this information quickly, an initial high-level model was developed using a project scheduling approach. This model represents a rough-cut analysis tool that eventually will be augmented by a second, detailed high-level model for a more comprehensive view of the Ingalls manufacturing environment down to and including the shop floor.

7.1 Project Scheduling Approach

Johnson and Montgomery [1974] distinguish project scheduling from job shop and other related types of scheduling by the nonrepetitive nature of the work. Several techniques have been employed to assist in the scheduling of projects. Critical Path Method (CPM) and Program Evaluation and Review Technique (PERT) are two historically popular methods for this purpose. Although both techniques are similar, PERT attempts to account for uncertainty through the use of probability distributions. Generally, this takes the form of a beta distribution unless historical information is available to provide a more appropriate representation.

While these techniques provided an improved tool for project scheduling, they have their limitations as well. Although simulation modeling overcomes many of these, the precision of its solution is a function of the level of detail used in the design of the model. Since detailed models require detailed data, a potential limitation could exist with the use of simulation if the data collection process proves difficult or time consuming.

A simulation model that provides less precise "rough cut" approximations of schedule performance can be built quickly though with two simplifying assumptions. First, by representing only major activities in the model, a reduction in detail can be achieved. Second, where historical data is not available, use of a unimodel distribution such as the beta distribution, provides an approximation until additional data can be obtained. To provide a forecasting tool to management as quickly as possible, an initial high-level model based on these assumptions was built.

7.2 Initial High-Level Model Architecture

Schedule input to the high-level model is accomplished through the same mechanisms that are used by the low-level model. Because each assembly work order package number includes a work center identification code as part of its structure, matching of bill numbers to the appropriate hull assembly data file is easily accomplished. Once the entity has been created, it is inserted into a station representing the input point for its work center.

Work centers in the initial high-level model are represented simply as a series of resource contentions for manpower and storage space. Provisions for intra-work center material handling, route sequencing, and machinery are not included at this level. Each entity entering a work center is assumed to require the same resource type and sequence. For example, all assembly work order entities entering the fabrication work center seize six resource categories to complete processing. These resource categories represent the six manpower crafts (fitters, welders, chippers, straighteners, shapers, and layout) utilized during the fabrication process.

Actual assembly work order package processing within a work center occurs as follows. Upon arrival at a work center, the assembly work order package entity is duplicated into several daughter entities. These entities are then placed into LVF (due date) disciplined queues for each of the resource categories represented in that work center. When released by these queues, each entity will attempt to seize units of the resource they required. The number of units seized is calculated as a function of the current simulation time and the due date of the entity. If insufficient units are available, the entity will seize only those units that are free--no preemption will occur. Once the resource allocation process is completed, an attribute assignment is made to the entity to record the number of resource units seized. This assignment is necessary for subsequent computation of processing delay and for resource release following process termination.

The total delay time required for processing is based on the number of resource units seized (manpower assigned) and the total number of man-hours required for processing. Man-hours required for processing is computed as a percentage of the standard work hours assigned to the parent assembly work order package entity when it was first created by Event One. Since each child entity has the same attributes of its parent entity, a man-hours calculation can be made for each resource category. For example, if the percentage of standard work hours required for welder resource processing is 75 percent, and the standard work hours assigned is 400 man-hours, then 300 man-hours are required for welder processing by the child entity generated for that resource category.

Once the number of man-hours required for processing has been calculated, the total processing delay time for that resource category is then determined. By dividing man-hours required (or remaining) by number of resource units allocated, a total delay time in hours is generated. Although a single SIMAN delay based on this time would result in completion of processing requirements of this entity, this approach would allow no additional manpower assignments until completion of the first delay--when the requirement no longer existed. This strategy could result in longer processing delays than necessary due to insufficient manpower assignments. In addition, it would require all other entities arriving after the resource pool began processing to wait for a delay completion to obtain free units. If this wait is excessive, due dates for these arriving entities may be missed.

To assist in preventing this problem, a "time slice" mechanism is used. Rather than delaying for the total processing time required, a delay time equal to the smaller of four hours or the remaining processing time is used. This strategy forces the entity to release its resources and compete again for resource allocation.

Upon completion of the processing delay, the amount of remaining delay time is determined. If the value calculated is non-zero, the remaining time is assigned to an attribute for subsequent use. This entity will require an additional delay cycle(s) before it has completed the processing requirements for that resource category. However, if the value calculated is zero, then the entity is considered to have completed the processing requirements for that resource category. This completed entity will release its resource units and move forward into a SIMAN "matching block" to wait for its sibling entities. When all sibling entities for the parent assembly work order package have finished processing at their respective resource categories, a single entity is released representing the completed assembly work order package. If work-in-process storage space is available, this completed entity will move forward to an intermediate queue for use by the next work station.

For child entities with processing time remaining after their delay cycle has been completed, a second duplication occurs. The primary entity is first returned to the original input queue for its resource category. Once this entity is in queue, the second entity is used to release the resource units originally seized by the primary entity. Following this release, the second entity is disposed. Since this release occurs after the primary entity has returned to the input queue, the primary entity is in position to again compete for the released resources based on due date. This cycle continues for all resource categories within that work center until the original assembly work order package has completed processing.

Downstream work centers requiring the presence of completed assembly work order packages from other work centers check appropriate inter-work center queues for their existence. This is accomplished by examining the bill number attribute value of each entity resident in these queues. If the required entity is found, it is removed from the queue for disposal; and the new assembly work order package begins processing. If the required entity is not found, the new assembly work order package enters a detached queue and waits for a signal from the required entity. This signal is sent when the completed assembly work order package entity arrives at the intermediate queue. This assemblage process continues until a single entity, representing the completed ship, has been produced.

Several statistics are maintained throughout the high-level model. At the work station level, time between assembly work order package exits, package throughput time, number of packages completed early and late, manpower utilization, and storage space utilization are but a few. At the shipyard level, inter-shop storage space utilization, time-in-storage, and ship throughput times are obtained.

8 SUMMARY AND CONCLUSIONS

This paper has described a bi-level approach to modeling the complex operations involved in shipyard operations. It has demonstrated that simulation models can be developed for large, complex manufacturing operations in considerable detail while maintaining a high degree of modularity in their design. The techniques used to design and implement the simulation models were presented; they were the project management, data analysis, the FORTRAN schedule interface routines and the low- and high-level SIMAN models.

While this approach was limited to the shipbuilding industry, the concept of parallel modeling efforts at two levels of detail can be applied to any environment requiring multiple views of the same system.

ACKNOWLEDGEMENTS

The authors extend their appreciation and thanks to their fellow team members: Roger Aucoin, Lisa Breland, Todd Buckley, Bill Davis, Lisa Greenberg, Darin Inlow, and Jimmy Taggart. The authors would also like to thank Ron Maze (Manager, Industrial Engineering, Ingalls) for his support of the project, Randy Sadowski (V.P., Services Division, Systems Modeling) for his help with the project in all its phases, Mary Newsome for proofreading, and Lynn Barrett for final preparation of this paper.

REFERENCES

Bergstrom, R.P. (1991) "The Hour When The Ship Comes In," *Production*, Vol. 103, No. 8, pp 44-51

Johnson, L.A. and D.C. Montgomery (1974) *Operations Research in Production Planning, Scheduling, and Inventory Control*, J. Wiley and Sons, New York, NY

Pegden, C.D., R.E. Shannon, and R.P. Sadowski (1990), *Introduction to Simulation Using SIMAN*, McGraw-Hill, Hightstown, NJ

AUTHOR BIOGRAPHIES

MARK K. BRAZIER is an engineer at Systems Modeling Corporation. He received his B.S. degree in Computer Science in 1977 and his M.S. degree in Industrial Engineering in 1987 from Texas A&M University. His responsibilities include performing manufacturing modeling, analysis, and training functions for the simulation services division. His areas of interest are in manufacturing and production systems including modeling, analysis, inventory control, and applied scheduling. He is a member of AIIE, SCS, and APICS.

CARL J. PERRY is currently the supervisor of Performance Analysis and Simulation Modeling at Ingalls Shipbuilding. He received his bachelors degree in Industrial Engineering from the University of Florida. His responsibilities include supervising the development and use of specialized computer programs for the evaluation of production performance levels, manning requirements, schedule position, and ship to ship learning curve comparisons.