

## COMMON FEATURES OF SIMULATION BASED SCHEDULING

F. Paul Wyman

Wyman Associates, Inc.  
181 Second Avenue Suite 321  
San Mateo, California 94401

### ABSTRACT

In recent years, "Finite Scheduling" and "Simulation Scheduling" have become popular as manufacturing software applications. From our communications with literally hundreds of companies over an eight-year period, we have prepared a list of the most frequently requested capabilities for simulation based scheduling, across a broad spectrum of manufacturing industries. This paper presents those features.

### 1 INTRODUCTION

This paper presents a set of features which customers expect and vendors are starting to provide in Simulation Based Scheduling (SBS) systems. SBS systems are a "precision" form of a more generic category popularly known as "Finite Capacity Scheduling" (FCS) systems. We provide a short history to explain this terminology.

Most companies do require additional items peculiar and customized to their own environment. This paper is not an exhaustive list of all features desirable in simulation-based scheduling.

A broad range of manufacturing companies are recognizing SBS systems as both practical and effective finite scheduling systems. This range includes metal-working job shops, pharmaceuticals, plastics, electrical, electronics, and airlines.

This paper is impressionistic in nature, and is not based on a formal statistical survey, but it is confirmed by 50 person-years of developing and promoting software products in the field. The remainder of this paper is organized as follows:

- History and Background of Simulation Based Scheduling
- Finite vs Infinite Scheduling
- Basic Concepts of Simulation Based Finite Scheduling
- Primary Features

- Intermediate Features
- Advanced and Special Features

NOTE TO READER: A graphic slide accompanies each paragraph in sections 3, 4, 5, 6 and 7. At the session, hardcopy of the graphics was distributed due to limitations on paper length. Readers may request a copy of the accompanying slides from the author.

### 2 HISTORY AND BACKGROUND OF SIMULATION BASED SCHEDULING

To our knowledge, the earliest Simulation Based Scheduling (SBS) system was pioneered at the El Segundo Plant of the Hughes Aircraft company in the years 1958-1964 as referenced in LeGrande (1963), Steinhoff (1964), and Bulkin et al. (1965).

The Hughes Aircraft computer for that application was an IBM 1410 with 80K words of memory, capable of scheduling only 160 of some 2000 jobs currently active on the shop floor. The author recently contacted the El Segundo Plant, but could not locate any person aware of this early application nor the whereabouts of the principal investigators. Simulation Based Scheduling had to wait fully two decades to become a commercial "off-the-shelf" software application with recognizable features and benefits.

In the 1960's, manufacturing software advanced under the "MRP movement." The word "finite" stems from the recognition that resources and capacity are limited, as opposed to many MRP-based scheduling systems which over-simplify capacity constraints, neglect realistic details such as queuing and capacity that changes substantially by work shift.

But finite scheduling methods in that day were crude, since they were based on what we call the "bucket method table." This table is conceptually similar to a large FORTRAN array or more simply a spreadsheet. The table represents shop capacity over time, where each row is a resource, each column is a time interval,

and cells represent either hours used or hours un-used. The finest time bucket is typically a single day. This approach typically glosses over the details of task queuing and contention that occur within the time bucket. Errors accumulate and result in non-feasible schedules, even when the time bucket is to the hour.

In the 1970's, finite scheduling was criticized as adding complexity and updating burden without producing sensible shop schedules realistic enough to be followed by the shop. Parallel with this development, other "Finite Scheduling" systems evolved based on linear-programming or having an "optimization" flavor. They were intended for assignment of products to plants or to capacity planning with time buckets of weeks to months. Other methods optimized setup sequences in order to maximize the utilization of "bottleneck" work centers to improve shop throughput.

Since 1983, our company (among others) has been developing and distributing production scheduling software based on discrete event simulation. Our own product (JobTime Plus), is implemented using the commonly know dBASE programming language for PC-compatible computers. The event list and queue priority features of discrete event simulation were implemented using indexed sequential concepts identified in an earlier paper by Wyman (1975). We also plan to offer C-based products in 1992 for the UNIX and VMS platforms.

Other systems reported as SBS include the product FACTOR from Pritsker Corporation, CMS from Manufacturing Management Systems, Schedulex from Numetrix and PROVISIO from AT&T Istel. There may be other SBS products on the market of which we are not aware.

Our original "Simulation Engine" was the generic job shop queuing simulator so familiar in introductory courses on simulation. But we rapidly learned that manufacturing companies need a far richer set of modeling features to express realistic details in order to create schedules that can be followed accurately on the shop floor. Our first system shipped in September, 1984, and we have added improvements continuously at the request of our clients.

There is growth in SBS systems commercially available. In 1984, we knew of five systems offering FCS and one SBS. Today we see nearly fifty commercial FCS systems, with five identified as SBS. Several additional systems are under development.

Only the SBS systems qualify as "Precision Finite-Capacity Scheduling" systems because they not only recognize finite capacity, but they also add the power of simulation to reflect realistic constraints.

A qualified SBS should be able to calculate capacity utilization accurate to the minute for each resource, including start and finish times of each task through each work center, while considering variable capacity over

time, flexible routings, setup-matching, parallel required resources and release-date tuning. The SBS should also be useful as either a simulation study tool, or for detailed short-interval scheduling for shop floor control. Flexible graphics and queries are essential.

### 3 FINITE VS. INFINITE CAPACITY SCHEDULING

#### 3.1 WC Load Status Illustration

It is challenging to visualize the distinctions between various forms of scheduling. To illustrate, we consider a simple example and show the differences in each scheduling method.

##### 3.1.1 Task Arrivals and Duration

First assume you have five work orders, with a total of eight hours of working capacity. But due to upstream work in other work centers (or other reasons such as move-times or material staging), only 2 jobs are on hand at 8:00 a.m. 2 more jobs will arrive at 12:30 p.m., and one more will arrive at 2:00 p.m. We assume no lunch break for simplicity, so that an eight-hour day starts at 8:00 a.m. and ends at 4:00 p.m. The work center has a single machine available.

##### 3.1.2 Finite Forward Pass

The finite forward (simulation) solution will work on one job after another, with queuing at 8:00 a.m. and 12:30 p.m. and a period of idle time from 9:30 a.m. to 12:30 p.m. The eight hours of work are not finished by 4:00 p.m., and the last job is not finished until the following day at 10:00 a.m. We present simulation first because it is closest to reality and easiest to understand.

##### 3.1.3 The Bucket Method

Next we see the bucket method of finite scheduling based on a daily time bucket of 8 hours of capacity. This method neglects the queuing and movement between work centers within the day. It has to rely on "approximation," assuming that the "small" details within the day are errors that will cancel each other out. Therefore the Bucket Method sees 8 hours of demand and 8 hours of capacity and proceeds to report all five jobs as starting and finishing on the same day. (Time of day is not computed or reported). Hourly time buckets have less error, but errors are still present and tend to accumulate in an realistic size schedule with a substantial number of jobs and tasks.

### 3.1.4 Infinite Backward Pass

Third, we see the traditional MRP “Backward Pass” approach. The jobs are assumed to finish on their due date, with start times equal to due date minus processing time. Each job is scheduled as if it were the only job in the shop. The capacity consumed by other jobs in the same work center on the same day is ignored. The implied behavior of the system is that the machine is idle until the very end of the day and then suddenly can do the work of five machines during the last few hours of the day. (Any relation to your own shop is purely coincidental). If the schedule system works like this, the shop tends to be forced to behave like this.

### 3.2 Adjusting the Backward Pass

To solve the problem with backward pass, its proponents adopted a “queuing allowance” as a buffer of time between consecutive tasks. The actual queue time by work center is normally ignored in favor of a single “plug” number such as “1 week,” regardless of the queuing conditions at individual work centers. This creates enormous “Q-Ratios” of Cycle time divided by work content of 20 or greater. This leads to significantly and unnecessarily high WIP (Work In Process) inventory.

### 3.3 Load Pattern for Infinite Capacity

The load pattern for infinite capacity is back-loaded. Capacity is severely overloaded just at due date and underloaded in the short term. Infinite forward loading would produce a highly overloaded capacity in the short term and underload in the long term.

### 3.4 Load Pattern for Bucket Method

The bucket method improves on Infinite Capacity by front loading capacity. It is less overloaded up front than the Infinite Capacity solution, but it is still loading work higher than capacity because it neglects queuing and other contentions inside each time bucket, i.e. it tends to over-estimate capacity

### 3.5 Load Pattern for Simulation Finite Forward Pass

Finite scheduling based on simulation does not overload capacity up front. The load tapers off eventually converging to distant future demand, but in the pattern most consistent with the realistic capacity constraints and shop floor conditions.

### 3.6 Divergent Comparisons

Errors in a stochastic, autocorrelated system do not simply cancel each other out. Instead the errors accumulate and amplify erroneous behavior over time. The greater the error in assumptions, the greater the divergence from reality, and from steady state behavior. The bucket method, and other “over-simplified” finite schedulers are less divergent than Infinite Capacity, but only accurate finite scheduling can produce stable behavior. We maintain that simulation based scheduling is the technology that supports the greatest achievability of accurate finite scheduling.

### 3.7 MRP vs. PRM

The concept of PRM is MRP in reverse. Precision Requirements for Materials helps MRP realize its goals of efficient materials management. In order for this to happen MRP needs to accept input from the shop floor. The shop floor simulation based schedule defines exactly when materials will be required, instead of the reverse flow of information. Many companies feel they now have more control over their vendors and purchasing (due to MRP) than they have over the capacity on their own shop floor. SBS provides an opportunity to improve the performance of the MRP system.

## 4 CONCEPTS OF SIMULATION BASED FINITE SCHEDULING

### 4.1 Capacity - Work Centers

Capacity must be recognized as the resources that constrain production whether they be machines, people, tooling, rooms, hallways or materials handling equipment. Such resources may not be included in a manufacturing system such as MRP. It should be possible to visualize work centers generically as “resource groupings.”

### 4.2 Capacity - Calendars

The plant working calendar should recognize variable resource levels by shift for each work center. Holidays and shift starting time are also important. To measure capacity accurately, it may be necessary to specify shift patterns that vary during the week and have irregular starting times and shift length by day of the week.

### 4.3 Jobs and Tasks

Each job order or work order consists of several tasks or operations. For example the “job” record is a master

record with “detail” records for each operation to be performed.

#### 4.3 Standard Routings

The relation of jobs and tasks continues with the concept of a standard routing for a standard part. The standard routing is a sort of “template,” containing standard work centers, setup time, processing rates and preferred work stations, setup codes, resources required and move times. The template can be copied repeatedly to create actual jobs.

#### 4.4 Job Specific Detail

Once an actual job is created, certain items will need to be added to identify this particular job, such as customer ID, work order number, part number, due date, release date and quantity. Depending on your modeling of jobs, additional data may be added to operations of actual jobs, such as a release date and time of day on a specific operation. Other adjustments should be allowed to default values from the template such as scrap allowance, labor cost rate, or any other field.

#### 4.5 System Constants

The user needs to set system constants such as the schedule start date, the standard number of minutes per shift, whether process time is units/hour or hours/unit and background constants controlling printing and other housekeeping conditions of the system.

#### 4.6 Scheduling Phases

Once the system is installed and loaded with data, a regular cycle is performed repeatedly, probably weekly or twice-weekly to reproduce the next detailed shop schedule. We believe it is unwise to reschedule too frequently such as daily or hourly. Meetings with materials and engineering help tentative “trial” schedules evolve into the next firm schedule, which then affects all related planning.

### 5 PRIMARY FEATURES

#### 5.1 Sequential & Positive Move-Time

The move-time field is normally seen as having value of either zero or a positive number, representing the time to move to the next work center. For example a move time of 0.25 means allow one quarter hour (15 minutes) after the current task is completed before starting the next task.

#### 5.2 Sub Assembly-Assembly

Jobs should allow the definition of assembly relationships to any limit to mirror the bill of materials relations between work orders. The system should not schedule assembly until all work orders representing the component parts are finished.

#### 5.3 Batch-Lot Relationship

Frequently a large batch is made and quantities are earmarked as inputs to downstream work orders. The Batch Job/Lot Job relationship is necessary to prevent “lot jobs” from starting prior to a batch job which fabricates a common component which has been designated for use by multiple lot jobs.

#### 5.4 Copy Jobs Feature

Once a job structure of batches, lots and assemblies has been defined, it will frequently be desirable to add an identical series of the component jobs in order to reflect a whole new customer contract. The system should permit copying a range of jobs, complete with structural relations, scaling quantities and initializing customer, due date, release dates and quantities.

#### 5.5 Release Dates: Jobs & Tasks

The principal control the scheduler has over jobs is the release date. If a job is being started and finished 15 weeks ahead of its due date, then you are losing money from idle finished goods inventory. Release date lets jobs be released earlier or later as desired. At the task level, both release dates AND time-of-day are necessary to control operations within the day.

#### 5.6 Scheduling Rules

An SBS should have a healthy variety of scheduling rules as well as the ability to add custom rules. Different rules reflect various heuristics and goals. On a given mix of jobs, you should be able to try several rules and compare the overall effectiveness of the rules.

#### 5.7 Score Card Reports

A score card reports how well your scheduling system is doing, particularly in regard to key parameters such as on-time performance, utilization, queuing and cycle time, and variability of performance. The score card should compare rules and “what-if” conditions.

## 5.8 Reports On Screen & Hard Copy

An SBS system should provide a broad variety of reports which demonstrate the kind of data that can be extracted from the scheduling database. Key reports are a work center to-do list (dispatch list) and capacity utilization over time to assess loading conditions. The system should include a report writer powerful enough to enable a non-programmer to create and add custom reports to system menus without technical assistance. It should be possible to preview each report on screen before printing to paper. Ad hoc queries are essential.

## 5.9 On-Screen Graphics and Reports

On-screen graphics help the user display utilization issues visibly for management and the shop floor. Larger issues such as bottlenecks and projected trends in cycle-time also are effective as presentation graphs. The system should allow you to add graphics readily, either inside or outside the system, and to print them on a variety of printers, plotters, or other devices.

## 5.10 Updating Methods

Updating can take place on various levels. The simplest is to manually update tasks fully or partially complete. Shop floor tracking is an alternative, as is downloading from an MRP system. Once schedules are being followed accurately, another alternative is "Automatic Updating," discussed below.

## 5.11 International Date Formats

The American convention of presenting dates as Month-Day-Year is not comfortable to much of the rest of the world. A flexible system should permit the user to select other date formats such as ANSI (YY.MM.DD), English (DD/MM/YY), German (DD.MM.YY), or Italian (DD-MM-YY).

# 6 INTERMEDIATE FEATURES

## 6.1 User Defined and JiT Overlapping

Job cycle time can be significantly reduced if material can directly flow from one work center to the next. Obviously this is not practical if your scheduling system cannot reflect the concept of overlapping. Special negative move-times allow tasks to overlap. This supports the JiT concept of manufacturing. JiT is supported even more if the SBS provides special codes which, if selected, will automatically perform the tedious calculation of move-

times synchronized to consider setup times and processing time/unit for the two overlapping tasks.

## 6.2 Concurrent & Parallel Operations

An additional modeling concept should support parallel concurrent operations (such as a parallel start of two or more unrelated tasks). An additional powerful concept is to use multiple tasks to reflect the requirement of multiple simultaneous resources. For example, an operation requires 1 assembly line, 3 mechanics, 17 assemblers, and 4 service workers.

## 6.3 Automatic Updating

Automatic updating requires that the schedule is being followed faithfully by the shop, and that all distracting forces have been controlled to the degree that schedules are followed closely. Automatic updating means that the system automatically updates itself, assuming that work was actually performed as scheduled, and the user marks the exceptions. We have seen several companies achieve 95% schedule accuracy in less than six months. This feature is practical, achievable, and saves enormous updating effort.

## 6.4 Max-Splits Feature

The SBS should allow the user to designate automatic lot splitting in which a large lot is assigned to several idle resources in a work center with surplus capacity, which would otherwise be left idle. There may be 15 spindles available and a task may designate that the maximum number of splits for that task is 5. Up to 5 spindles would be assigned to the job, incurring additional setups on 4 spindles, but substantially reducing the job's cycle time.

## 6.5 Minimize Setups

Matching setups is very important to manufacturing customers. This can be done by reflecting a matrix of tradeoffs or by designating a matching concept based on group-technology. Attempting to look downstream and optimize the entire plant around an single work center schedule is usually impractical, due to the random events impacting the schedule. It is more practical to dynamically select the best matching setup, classifying a degree of match such as major, minor or very minor, based on the set of jobs currently in queue or in transit to the work center.

## 6.6 Priority vs. In-Process Status

Shops that interrupt and preempt jobs in process will have more jobs in process than available machines as the SBS loads the shop. Users vary in their preference of allowing in-process work to override job priority level or vice versa. The SBS should let the user choose.

## 6.7 Milestones on Tasks

Many jobs have slightly different routings but pass through common work centers representing "milestone" events such as "Furnace" or "Q/C inspect." Sales and customer support require the ability to track common milestones over a variety of jobs, to track, gauge and report general progress on jobs. This is especially true for jobs that spend several weeks in process.

## 6.8 Early-Late Cost Penalty

The ideal scheduling system would make each job finish exactly on time, starting as late as possible, still keeping capacity fully utilized. Days early are undesirable as are days late, but the penalties are different in nature. Therefore it is desirable to track two kinds of data: number of jobs early, average earliness and maximum earliness; number of jobs late, average lateness and maximum lateness. The penalty for lateness is normally greater than earliness. A weighted penalty measure can be used to compare quality of alternate schedules.

## 6.9 Automatic Release Date Adjustment

Once a simulation schedule has run, it is desirable to be able to "smooth" job release dates so as to maximize on-time performance. Jobs that are early can be released later, by a portion of their earliness. Jobs that are late should be released earlier, if possible, by at least as much as their lateness. In this way, iterative schedules converge to a "near-optimum" with respect to on-time performance. This is a heuristic, not optimization.

## 6.10 Flexible WC Capacity Analysis

The typical analysis for scheduling is the utilization of work centers over time. The user checks closely for under or over utilization, as well as un-used capacity when quoting new contracts. A flexible SBS will display by work center or individual resource, adjustable time intervals (shift, day, week, month) and variable number of intervals (e.g. 12, 20, 60). This flexibility should apply not only to summary statistics for all tasks processed at the work center, but also show individual tasks over time in Gantt-style format by bars, hours or headcount re-

quired. Graphical as well as tabular display is also essential for this type of analysis.

## 7 ADVANCED AND SPECIAL FEATURES

### 7.1 Maximum Overlapping Jobs

Job shops can achieve JiT performance if they schedule properly to allow sub-jobs to be synchronized if the top-level assembly jobs can be fed continuously from all relevant sub-jobs. The SBS needs to assure that the top-level job is started as early as possible, but not until output starts to be available from all of the feeder sub-jobs.

### 7.2 Preferred Work Stations

This feature allows the user to specify on a given a preference order for selecting certain work stations within a work center, and also being limited to a given sub-set of the work stations. This feature is analogous to the feature of "alternate work centers" within standard routings.

### 7.3 Variable Capacity Over Time

Users need to adjust shift capacity of a given work center in order to reflect vacations, variable demand or planned maintenance. The calendar and capacity model of the SBS should support at least this degree of flexibility. This feature can be strengthened if the user can also specify shift length by day of the week, including start time and duration of each shift.

### 7.4 Seize and Release

Some operations need to seize a resource and hold that resource during a certain number of successive tasks. For example, you may want to have a limited number of molds, tooling sets or carts. You want to commit one of those resources to a given work order, and keep it committed until a certain number of additions tasks have been performed, and then release the seized resource(s).

### 7.5 First Start/Last Start

A certain work center may be open until 2:00 a.m. But you know that the current task of two hours MUST immediately follow work done on this work order in the preceding work center. For example, packaging immediately follows manufacturing a food product. Therefore you want to stipulate a last-possible start time limit on the manufacturing task to assure that it can be packaged

by the end of the day. Similar reasoning applies to using a "first-start" time constraint.

### 7.6 Batch Process Features

Process manufacturing requires that several work orders will occupy a common resource simultaneously. An example is batching orders of metal parts for heat treatment in an oven at a common temperature. Other processes are work order specific. Process treatment is highly company specific and the SBS should enable new logic to be added to reflect realistic requirements, including the overlap of pre-batch and post-batch tasks.

### 7.7 Exclusive Processes

When work center ABC is running process X, work center DEF cannot run process Y. A practical example is to assure that when gray paint is being used in work center ABC, work center DEF does not use bronze paint. (The paint released into the air will settle on drying parts from the opposite work center). An SBS should be able to reflect exclusive processes analogous to the linear programming constraint of  $X + Y = 1$ . Parallel multiple resource capability supports this kind of modeling in an SBS.

### 7.8 Tracking Interface

An SBS should support an interface with popular tracking systems that deliver data on task completion status. The typical data elements include work order number, date and time, quantity complete, hours worked and whether reporting is for setup or processing.

### 7.9 MF/MRP Interface

Most companies have an existing manufacturing (MF) or MRP system. The SBS should be able to download selected work order transactions, convert them to the SBS format, supplement data fields unique to the SBS and check for inconsistent input data. To get full benefit, the SBS should also upload scheduled work order start and finish dates to the MRP. The MRP should use the refined schedule dates to adjust its estimated demand and deliveries of material requirements. The SBS can help the MRP be far more accurate and therefore achieve JiT performance, even in random routing job shops.

## 8 SUMMARY

We summarize by suggesting that the features presented in this paper be considered a minimum set of capabilities that will be required in both FCS and SBS systems in

order to attract and hold the attention of today's manufacturing companies.

## REFERENCES

- Bulkin, M.H., J.L. Colley, and H.W. Steinhoff, Jr. 1965. Load forecasting, priority sequencing, and simulation in a job shop control system. Presented at ORSA/TIMS Joint Meeting, Los Angeles, 1965.
- LeGrande, E. 1963. The development of a factory simulation system using actual operating data. *Management Technology*, Volume III, No. 1, May, 1963.
- Steinhoff, H. W. 1964. Daily system for sequencing orders in a large-scale job shop. Presented at the Sixth Annual ORSA/TIMS Joint Western Regional Meeting, Orcas Island, WA, April 16-17, 1964.
- Wyman, F.P. 1975. Mechanisms to improve event scanning in discrete state simulation. *Communications of the ACM*, June 1975.

## AUTHOR BIOGRAPHY

F. PAUL WYMAN is President of Wyman Associates, Inc., and has developed and installed simulation based scheduling systems in over 50 companies since 1984. He holds a Ph.D. in Business Administration and has taught simulation in many universities, wrote one book and several scholarly articles on simulation, and won the 1977 ORSA/TIMS Edleman Prize for simulation practice on the Syncrude Tarsands Project. He has held management software positions at Bechtel, and was an early employee at Apple Computer and SBT Corporation.