JIT approach to integrating production order scheduling and production activity control

PRASHANT BANERJEE and HUSSEIN A ARMOUTI*

Abstract: Two of the most desirable features in integrating production order scheduling and production activity control is sustaining a finite workcentre capacity assumption for accurate production order scheduling, and efficiently performing a what-if sensitivity analysis to handle the dynamic uncertainties in production activity control. A finite workcentre capacity assumption leads to too many constraints that tends to slow down the computation process, and a what-if analysis tends to put too much strain on the design of the database system for production activity control. With a finite workcentre capacity and what-if sensitivity analysis capability as the basis, a holistic approach towards production planning and control is emphasized in this paper by simplifying the system based on JIT principles such as shallow levels in bill of material, a pull type backward planning and a clear identification of the type of operational environment for which the designed system is readily applicable, and data integrity by tight integration of the various modules in production planning and control. As an illustration, a system is designed and has been successfully applied to a rapidly changing made-to-order printed circuit board manufacturing environment.

Keywords: JIT, production planning and control, MRP, bill of materials, what-if-analysis

A just-in-time (JIT) approach to manufacturing emphasizes systematic identification and removal of wastage in the production process. Six such wastages are commonly addressed: overproduction, waiting time for products and operators, moving time for products and operators, poor manufacturing processes (including breakdowns), inventory, and scrap production[1]. The JIT approach emphasizes clustering of sets of micro level decisions, whereas another approach which is frequently contrasted with it, manufacturing resource planning (MRP II), keeps track of such relatively micro level decisions (see Flapper *et al.*[2] for a recent comparison of the two approaches). A comparison of records between JIT and MRP II indicates that JIT relies on omitting intermediate and relatively unimportant records and clustering them into fewer records, while emphasizing product quality improvement and production streamlining measures more. Omission of such intermediate details requires JIT to maintain a high degree of discipline and coordination among the production processes and also with the suppliers; otherwise, omission of the details would play havoc in terms of inaccuracies between what happens in actual production and what JIT maintains as it records the production process. Because of its emphasis on the current detailed records of the manufacturing process such as lead time, scrap rate, set-up time, required labour time, queue time, etc., the MRP II is more geared towards planning and controlling production activities, whereas the focusing of JIT towards wastage removal makes it valuable for streamlining the existing production at the expense of some loss of planning and control[2]. Because of the requirement of high discipline, JIT has been applied mostly on a case by case basis, e.g. the Toyota Production System[3], for which it is best known, whereas the MRP II approach has emphasized on generalization of the production planning and control problem. In this paper the JIT approach has been merged with some of the MRP II principles, and the resultant approach has been found useful for application in a rapidly changing make-to-order manufacturing environment.

The production order scheduling task in a multi-product, multi-stage system is a link between the high level decisions such as production planning and low level decisions such as control of actual production activity. Production planning determines the manufacturing role of a company's strategic plan. It is often expressed by a master production schedule, which is a disaggregated version of the production plan and is a statement of end items or product options that the company will build in future[4]. Approaches have been designed for controlling the master production schedule (e.g. Sridharan and Berry[5]) for easing the task of

Department of Mechanical Engineering (M/C 251), University of Illinois, Chicago, Illinois 60680, USA
*Cimnet Systems Inc., 601 Oakmont Ln, Westmont, Illinois 60559, USA

0951-5240/92/040283-08 © 1992 Butterworth-Heinemann Ltd

production planning and control. In this paper it is assumed that the master production schedule is totally flexible and dictated by the market or pull conditions, and as such does not impose any prefixed or forecasting based production targets. The master production schedule feeds directly to the production order schedule which then determines the time phased plans for all parts and raw materials required to produce all the products in the master production schedule. This segment is commonly known as materials requirement planning (MRP). MRP does not explicitly address production system capacity[6–8]. The production order scheduling is the detailing of the material and capacity planning. The production order scheduling computes the labour or workcentre capacity required to manufacture all the parts. The production activity control establishes priorities for all work orders at each workcentre so that the orders can be properly scheduled. A sophisticated version of production activity control frequently takes into account the finite capacities of each workcentre and make use of actual (current) information, while simplified versions often assume infinite capacity at each workcentre and try to establish work order priorities by heuristics such as historical information (see Biggs[8], and Hoffman and Scudder[9] for more details).

In production order scheduling, the short-term matching of capacity to demand is one of the core problems. The master production schedule is a longer term statement of the demand. But the demand is always subject to fluctuation because of a number of internal (e.g. prolonged machine breakdowns) and external (e.g. market economy) conditions, which happen so suddenly that the master production schedule is unable to account for such sudden variations. To handle such short-term problems, the master production schedule is often operated differently under three environments: make-to-stock, which is based on freezing the master production schedule for a certain time fence; assemble-to-order, which is primarily based on partially freezing the master production schedule; and make-to-order, which is primarily based on a liquid master production schedule (see Vollmann *et al.*[4], chapter 10). The make-to-order production environment requires the tightest production activity control, because commitments are already made to the customers, orders booked and inability to meet deadlines may have more severe repercussions than the other environments. The make-to-order environment can be further classified into two categories: one in which the orders are fairly static for long periods of time, e.g. airplane industry; and one where the orders are dynamic and get updated frequently, e.g. printed circuit board (or printed wiring board) manufacturing industry. Such a classification of the manufacturing environment prior to the solution process appears to streamline the production order scheduling task, and consequently allows one to place more emphasis on a tight integration of low level production activity control decisions with production order scheduling, which may otherwise be progressively difficult because of the enormity of a more general problem formulation. The above observation is one of the off-shoots of JIT, which emphasizes a clear definition of an industry before attacking the scheduling and control problem. In our view, with the advent of JIT philosophy research should now be focusing more on environment specific requirements in designing production order schedules and exploring issues for its integration with other decisions in manufacturing systems. In this paper, these issues are systematically analysed for a rapidly changing make-to-order production environment.

The described interface of production order scheduling and production activity control under a dynamic make-to-order environment has been geared towards industries with relatively short manufacturing cycle times, that is, the time period from the receipt of order from the customer and the delivery of the finished product ranging from 2–3 days to 2–3 months. The production is based on a backward planning starting from the receipt of a work order. No make-to-stock and assemble-to-order environments are deemed necessary because of the rapidly updating work orders and short manufacturing cycle times. This assumption also precludes the possibility of accumulation of inventory because the control is from the pull side. A tight integration of all aspects of the production process is needed in successfully implementing the production planning and control for such a production environment. The previous history of JIT shows that it has been successful when cautiously applied on a case-by-case basis, and incorporation of this JIT philosophy with the sophisticated planning and control ability of an MRP II type approach on a case based production environment (which in this instance is a rapidly updating make-to-order environment with short manufacturing cycle times) promises to be a good platform for integration of production order scheduling and production activity control.

Motivation

The integration of production order scheduling with production activity control is emphasized for the following reasons:

Addressing the shop floor conditions in production order scheduling

A general production order scheduler is likely to assume infinite loading at each workcentre because it is only a system at the level of production activity control that is able to meaningfully assess the impact of finite loading at each workcentre, taking into account the existing limitations such as workcentre performance limitations and preventive maintenance down times.

Need for data integrity

Data integrity is an essential ingredient not only for integration of production order scheduling and production activity control, but also at a much more macro level involving integration of various modules in a company (e.g. finance, manufacturing, sales, purchasing and engineering). A poor data integrity leads to record inaccuracies because some units may not conform totally to a schedule, and may hence operate in their own isolated way according to their convenience without having an efficient channel for informing the other units. A survey of 326 companies using MRP[10,11] indicates a strong correlation between the degree of record accuracy and the degree of conformity to MRP II, e.g. in the form of financial and operating systems closely tied together along with what-if capacity[12].

With a number of work orders arriving frequently in the rapidly updating make-to-order production environment, the records have to be updated frequently, and the absence of a strong degree of data integrity may lead to a lot of record inaccuracies.

Finite versus infinite loading, what-if capabilities

Finite loading systems simulate actual work order starting and stopping to produce a detailed schedule for each workcentre by including all the jobs in all the required workcentres for the length of the planning horizon, assuming finite capacity limits at each workcentre. A finite loading system has the advantage that it is able to better forecast the actual load on each workcentre in the near future. For a finite loading system to be effective, very accurate standard time estimates, capacities and stringent data integrity measures are needed. Because of the frequent arrival of new or updated work orders into an almost static total production capacity that is available, the finite loading system has to be updated frequently to retain the accuracy of the schedules. The difficulty with this requirement is that with so many variables and constraints, the system itself takes some time to run, and it is expensive to frequently implement the updates indicated by each run. The challenge here is to design a system that is able to run efficiently. Many existing systems attempt to optimize themselves mostly by freezing the schedule for high priority items and then juggling with the schedule of the lower priority items. This procedure disturbs the optimality of the original results and gives us quick sub-optimal solutions. A better alternative, such as that reported in this paper, is a system that is re-run by erasing all current schedules. This is possible because of the steps taken to improve the run time efficiency. The most desirable alternative would be a system that is able to optimally update itself without re-running from the beginning, and this is a subject of future research.

A finite loading system can be contrasted with infinite loading which does not consider workcentre capacities. The selection of finite *versus* infinite loading is often based on a choice between handling too many or fewer loading constraints. While both have their advantages and disadvantages, a compromise solution has been provided in optimized production technology (OPT) approach[13]. In OPT, all jobs are sorted on the basis of usage or non-usage of certain bottleneck workcentres. A bottleneck workcentre is one that is loaded to a high level of its capacity. Small subsets of jobs that cross bottleneck workcentres are finitely loaded, whereas the majority of the jobs are scheduled using infinite loading. In this paper, finite loading is used as the default method, with infinite loading used to perform what-if sensitivity analysis in case of bottlenecks.

Efficient computer memory management

It has been generally acknowledged that the what-if sensitivity analysis for capacity planning creates severe demands on the database design[4]. The concept of bit-mapping has been used to efficiently manage the computer memory. Bit-mapping allows one to explicitly specify how every bit of computer memory is to be handled. This overrides the default allocation of bits by the computer, namely the use of regular memory buffer, e.g. the internal memory allocation for an array. The bit-mapping procedure addresses the speed and memory requirement by reducing overhead in data storage and retrieval.

The memory management can also be made more efficient by design, e.g. if the inventories are kept low, the bill of material levels are shallow, and lead times are short, then the what-if analysis can be relatively inexpensive[4]. These characteristics are used in the design of what-if capability, from a memory management standpoint.

Methodology

The adopted methodology is shown in Figure 1 in the form of a flow chart and explanation of the blocks. More detailed explanations are provided here.

Since the system is part of an integrated package that controls all the manufacturing activities, the package starts the cycle from the first activity, a request for a quote, to the last activity, shipping and payment collection. It helps the management to adopt the JIT philosophy of planning backwards with minimum cycle time from the customer order to the first production activity. The main goal of a manufacturing environment is to reduce cycle time, the work in process cycle is one of the most important cycles that we are trying to minimize, which is the goal of this system.

The system comes into the picture after the integrated package has finished creating the work orders based on their respective bill of material structure. For each finished item, and each subassembly (which is anything but raw materials) a work order is created.

The work orders are prioritized based on a set of user defined factors (this is also referred to as the sort order) which are: scheduled completion dates (due dates); priority (each work order is assigned a priority by the planner, although the exact reason for assigning a particular priority is hard to quantify; some of the reasons are urgency, importance to customers, and nature of workcentres through which the work order will travel; quantity scheduled; and work order entry (creation) dates. The bill of material is used to build a tree of work orders. Only the finished items are used in determining the priority of all the work order tree members. When the scheduler is fired, three data items are asked for cut off date: maximum number of days; and schedule effective date and time (the date and time that the scheduler should consider releasing the very first activity to each workcentre). The cut off date determines the upper bound on the duration of orders to be scheduled, if this date is a month from now, the scheduler picks the orders that are targeted to be finished within a month from now (the order finishing date is termed the *scheduled completion date*, and it is determined by human planners based on standard lead times). The maximum number of days is the upper limit which the scheduler should consider in determining the projected workcentre schedules; this should not be confused with the cut off date. When the scheduler tries to allocate busy resources against a work order, it pushes the booking to a later date to set a break line on

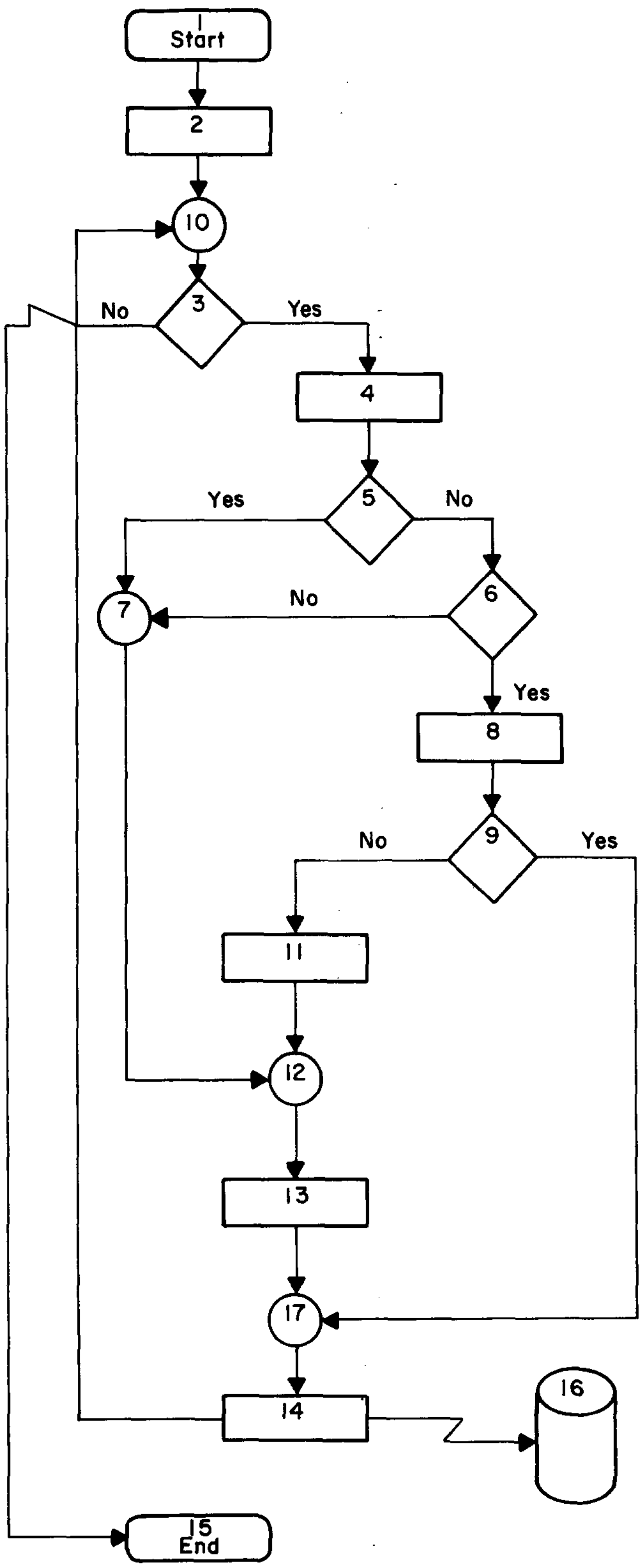
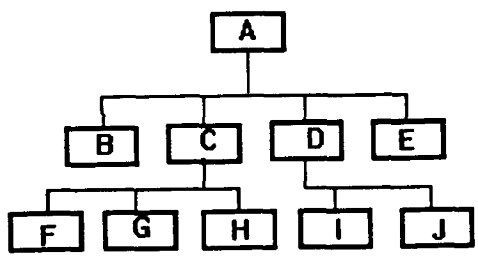


Figure 1. Algorithm for integrating production order scheduling and production activity control in a make-to-order production environment. 1. Start. 2. Prioritize work orders to be scheduled, i.e. select sort order. (Note: *Only finished items (i.e. top level)*) *of the bill of material are considered in determining priority. For example, node A in the diagram:*



3. Any more orders to schedule? 4. Consult bill of material to build a tree of work orders to be scheduled. The tree is a mirror image of the bill of material with each node corresponding to a work order to be scheduled to build that item. This structure implies that item D cannot be started before the completion of items I and J. 5. Has any work order in the tree been released *to shopfloor? 6. Is the Scheduled Completion Date (specified by management) ⩾ TODAY? 8. Try to backward schedule the entire work order tree starting at the top level and going down to lower levels. Try to meet the Scheduled Completion Date finish by starting at the Scheduled Completion Date and schedule backward in time and look for time slices (capacity available) at the workcentres of interest to come up with a Projected Start Date for this work order. 9. Was the backward attempt successful? 11. Clean-up schedules and prepare for the forward trial. 13. Try to forward schedule the entire work order tree starting at the lower levels and going up to top levels. At this point, it is known that the shop will not meet the work order Scheduled Completion Date, it is going to be late. 14. Store schedules and generate work order exception list (orders that could not be scheduled due to lack of capacity.)*

the later date booking; the scheduler takes the maximum days starting from the schedule effective date as its break line. If the scheduler cannot book resources against a work order within maximum days, we say the order does not fit.

The system starts by scheduling the orders based on their sort order. The order scheduled completion date is compared to the schedule effective date; if the scheduled completion date is later than the effective date, and none of the lower level work orders in the bill of material structure is released to shop floor, the order is backward scheduled, starting at the last step in the production route at the completion date and going backwards through the steps and time. this is done for the entire work order tree, starting at the finished item and going down the tree and backwards in time. A lower level item takes its scheduled completion date as its parent's start date. This ensures that no subassembly is released to shop floor unless its finished item will start in production immediately. If, while going backward in allocating resources, the scheduler comes across the effective date, then the order cannot be completed on time because no resources are available at this point; in other words, it does not fit.

If an order could not fit going backward from its scheduled completion date (the reason for this is lack of availability of resources), it is known at this point that the order will not be completed at its scheduled completion date; it is going to be late. The scheduler starts the forward scheduling process for this tree of work orders. The forward scheduling starts at the lower level items, starting at the schedule effective date and going forward in time allocating resources against the order. The finish date of the latest lower level items will be the start date for the higher items. If, while going forward, the maximum number of days is hit, we say that the order is partially scheduled, and the scheduler aborts this tree of orders and starts the same process again for the next tree of orders according to the sort criterion.

Having described the methodology in general, some of its specific features are highlighted.

User defined units

The above discussion assumed the availability of finite capacity resources. The capacity of workcentres is defined by 'user defined' units; for example, number of hits in a drilling machine, or time in another process. When defining capacities, engineers would set a certain level of capacity for all the workcentres available, e.g. drill one is capable of doing 500 000 hits per day, and will operate two shifts of eight hours each Monday through Friday, and one shift of six hours Saturday and Sunday.

Addressing shop floor conditions

The scheduler allocates resources against orders, and keeps track of when these resources will work on which orders, since the resources are finite. As soon the scheduler runs out of resources, for example, after allocating drill one for two shifts of eight hours each on Monday, the scheduler flags Monday as busy for drill one and never tries to allocate the drill on this Monday. It will start allocation on Tuesday. This approach reflects the real life case, and the scheduler considers the real capability of each of its resources and never overbooks the shop, hence management know what the shop floor will look like in the next 60 or 90 days (the maximum number of days). Management will also know which orders would be on time, and which would be late.

What-if analysis

To do what-if sensitivity analysis, the notion of workcentre utilization was introduced to the scheduler. Planners can elect to set an aggregate factor for each workcentre in each single day; this factor is the utilization factor. The scheduler takes this factor and multiplies it by the workcentre capacity to come up with another capacity which it considers instead of the original one, this way, the planner can set the utilization to a very high number, say 1000 and run the scheduler. The scheduler now will consider a capacity of 1000 times more than the real life capacity, in other words infinite. Planners can look at the shop floor picture for the next 60 or 90 days with the infinite capacity in mind. By the same token, the planner can set the utilization at a lower factor (less than one) and simulate a reduction in resources.

Highlights of interface between production order scheduling and production activity control

The above notion of utilization is used to alter the workcentre shift times. It lets the user define workcentre availability by the hour level. So the planner can start up and shut down a given workcentre using a given number of hours. The scheduler will not consider this workcentre to be available at these hours. This is also used by the plant maintenance system of the software package (details not included) to communicate with the sheduler to report the workcentre down times for maintenance so that the scheduler will not allocate any of these resources to any work order during down times. This also used by the statistical process control system of the software package (details not included) to trigger any workcentre going out of specification process control system of the software package to trigger any workcentre going out of specification limits that needs to be down for maintenance for a given period of time.

Bit mapping

The internal representation of resources is accomplished through a bit-map structure. For each workcentre each day, a 24 byte block is allocated; each byte represents one hour leading to a 7.5 minutes bit (one byte equals 8 bits). 7.5 minutes provides a reasonable grain or unit size for accounting for time in production activity control for a system which goes to as aggregate a level as production order scheduling up to the next 60–90 days. When the scheduler starts scheduling an order, it scans the bit-map for the next available bit (a bit can only be zero or one, 0 for available, and 1 for busy). The scheduler reads the database to decide the quantity to be scheduled in user units, and converts this to seconds. Since each bit represents 450 seconds, the scheduler knows how many seconds are needed at this workcentre, and tries to find a consecutive number of 0 valued bits to allocate; if so, it flips these bits to ones to reflect the allocation. The word *consecutive* is crucial, and implies that jobs cannot be split; in other words, if a process starts on a particular job it cannot be interrupted by another job. The reason behind this is the setup time wastage. Each time a process starts, it requires some setup time. If we keep interrupting this process, we keep setting it up over and over, wasting our resources on unneeded setups.

The capacity of a workcentre is measured in terms of the number of time slices, with 7.5 minutes per slice. Measuring capacity in terms of time-based units has the advantage of easily merging with many of the shop floor conditions, e.g. if a machine is down for four hours for maintenance, then 32 time slices will be deducted from the capacity of the machine for the scheduling time horizon under consideration. Similarly, setup times can also be deducted from the capacity of the machine. Thus allocating capacity in units of time provides a more uniform basis for integrating production order scheduling and production activity control.

The bit-map approach cuts down in memory requirement by the scheduler. The bit-map size is $W \times D \times 24$ bytes, where W = the number of workcentres, D = the number of days, and 24 = the number of hours. If the resources representation was in regular array structures, the array size needed will be:

- $(W \times D \times 24 \times 2)$ bytes if the array elements were integers
- $(W \times D \times 24 \times 4)$ bytes if the array elements were long integers or floating point numbers
- $(W \times D \times 24 \times 8)$ bytes if the array elements were floating point numbers with double precision.

This shows that the memory cut down is 12.5–50% of the memory requirements of the machine.

Example

The methodology described above has been implemented and applied to printed circuit board manufacturing. The rapidly changing make-to-order production environment includes, on average, about 200–300 finished item work orders, which are promised to customers, with on average 20–30 finished item orders or updated work orders being added every day. The output of the algorithm (outlined in Figure 1) goes into a database, from which output reports are generated. The database is relational, with about 200 data files, and is supported by Btrieve and SQL (Structured Query Language) calls. The PC communicates to a Novell local area network (LAN) which maintains the database. The run time of the software is about 40 minutes to generate the production order schedules, along with the production activity control instructions. The run time is a critical parameter which determines the success of the described methodology. To give an indication, a previous version of the methodolgy used to take about four hours to run, and this was considered too slow when the software was delivered to PCB manufacturers. (It takes about two hours to print the output reports on a line printer and about half an hour to distribute them on the shop floor).

Figure 2 illustrates the workcentre data which is supplied by the user. There are three levels in the hierarchy in the organization of each department. Each department is made up of one or more groups, and each group is made up of one or more workcentres. The user has the option to specify that work order has to be routed either through a particular workcentre, through a particular group, or through a particular department. If the user specifies that a work order is to be routed through a particular workcentre, then this is a very rigid specification and the algorithm does not have any freedom to route the work order through any other available workcentre for that particular activity in the department. On the other hand, if the user specifies that a work order is to be routed through a particular department, then the algorithm has the flexibility of routing the work order through any available group in the department, and through any available workcentre in that group. There are a number of other parameters which are specified by the user. Normally, a default value appears in the menu and the user has to specify only if he wishes to override the default values. The

```
Copyright (C) 1988-90                          Cimnet Systems, Inc.
10/ 7/91             SIVA'S DATA BASE       ID: ADMIN  A0415
                    PRODUCTION WORKCENTRE
                                                            (1)
WORKCENTRE CODE: DRL-W1-HD3          CAPACITY        : 20000.00
WORKCENTRE NAME: DRILLING M/C-3      THRESHOLD UNIT:
                 HEAD-#1
SUPERVISOR CODE: MAC                 LABOUR OVHD     :  10.00%
SUPERVISOR NAME: MAC                 MAT'L OVHD      :   3.00%
GROUP CODE     : DRL-G1-HD3          GEN'L OVHD      :   2.00%
GROUP NAME     : DRILLING M/C-3      FIXED BURDEN    :   0.00
                 HEAD
DEPARTMENT CODE: DRL-D               VAR BURDEN      :   0.00
DEPARTMENT NAME: DRILLING M/C        SCHEDULE        : Y (Y/N)
UNIT CODE      : HITS                REMOTE ENTRY    : N (Y/N)


SHIFT START STOP  RATE         DEFAULT STANDARD TIME(S)

 1:   07:00|15:00|11.00  PROCESS: 00:00:00:00 (DD:HH:MM:SS)
 2:   15:00|23:00|11.00  SETUP  : 00:00:00:00 (DD:HH:MM:SS)
 3:   23:00|07:00|11.50  QUEUE  : 00:00:00:00 (DD:HH:MM:SS)
```

Figure 2. User-specified parameters for each workcentre

parameters are self-explanatory, and hence are not described here.

Among many output reports, the most prominent are the projected workcentre schedules (see Figure 3), the dynamic workcentre loading report (see Figure 4) and the work order dynamic loading report. These reports are used by the user to prepare queries for what-if analysis.

Figure 3 illustrates the projected schedules for a drilling workcentre. The user-defined unit for capacity is the number of hits per day. The second column refers to the part number, which is also indexed along with its customer number (a part refers to generic class of items in printed circuit board terminology). Column 3 refers to the user-definable revision code of the work order, e.g. the original work order might be referred to as 'A', the next revision to it might be referred to as 'B', and so on. The fourth column refers to the work order number. The fifth column indicates the priority of the work order, e.g. the highest priority is indicated by 'A', the next highest priority by 'B' and so on. The priorities form the basis of the sort order, and the factors which determine the sort order, are mentioned in the methodology. The sixth column refers to the scheduled date or due date as desired by the customer for each work order. The seventh column refers to the projected starting date as determined by the algorithm. The eighth column refers to the starting time for the work order, with 0 indicating midnight and, for example, 430 indicating 4 hours and 30 minutes past midnight. The ninth column indicates the duration of the work order in hours and minutes, e.g. 1200 indicates 12 hours and 00 minutes. The tenth column indicates the routing level of the work order.

As discussed earlier, there are three levels of routing: department, group, and workcentre. The entry DRL-G1-HD3 in row 1 of column 10 indicates that the routing for this work order is at group G1 of department HD3. Columns 11, 12 and 13 indicate the same value in three different units. Column 11 indicates the current work order backlog in the number of parts (a unit specific to PCB manufacturing), column 12 indicates the current work order backlog in the number of hits (the user defined unit for capacity of this workcentre), and column 13 indicates the backlog in the number of panels (which is another unit specific to PCB manufacturing). Finally, column 14 indicates where the work order is currently located. If an entry say 'backlog', then that work order has not yet been started. Ideally, the column should always indicate the department, group or workcentre number which is currently being addressed; this means that the work order is on schedule. The projected workcentre schedules begin at midnight on any particular date, and

Copyright (C) 1988-1991 Cimnet Systems, Inc.
10/07/91 SIVA'S DATA BASE ID: ADMIN
16:04 Page: 1

PROJECTED WORKCENTRE SCHEDULES

WORKCENTRE NAME : DRILLING M/C-3 HEAD-#1 WORKCENTRE CODE : DRL-W1-HD3
SUPERVISOR NAME : JOHN MAC SUPERVISOR CODE : MAC
CAPACITY : 20000 (HITS) UNIT NAME : HITS DRILLED

1. CUSTOMER	2. CUSTOMER/INVENT PART	3. REV	4. WORK ORDER	5. P	6. SCH-DATE	7. PRJ-DATE	8. STRT	9. DRTION	10. RTE D/G/W	11. PRT-BLG	12. QTY-BLG	13. PANEL	14. LOCATION
CRP CUST	CRP PART - LEVEL 21		-000016-01-201	A	10/04/91	10/08/91	430	1200	DRL-G1-HD3	20	10000	20	SHR-G
CRP CUST	CRP PART - LEVEL 24		-000016-01-204	A	10/04/91	10/08/91	1630	1200	DRL-G1-HD3	20	10000	20	RMP-G
										40	20000	40	
CRP CUST	CRP PART - LEVEL 11		-000016-01-101	A	10/14/91	10/11/91	607	1200	DRL-G1-HD3	20	10000	20	Backlog
										20	10000	20	
									WORKCENTRE TOTAL:	60	30000	60	

Figure 3. Projected schedule for a drilling workcentre in output report format

Copyright (C) 1988-1991 Cimnet Systems, Inc.
10/07/91 SIVA'S DATA BASE ID: ADMIN
16:06 Page: 1

DYNAMIC WORKCENTRE LOADING

	BBL-W	BPL-W	CPR-W1-INL	CPR-W2-OTL	CUL-W1-REF	CUL-W2-TEF	DBR-W	DEV-W1-INL	DEV-W2-OTL	DRL-W1-HD3	DRL-W2-HD3
10/08/91	0.00%	0.00%	20.83%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	81.25%	68.75%
10/09/91	0.00%	0.00%	31.25%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	18.75%	31.25%
10/10/91	0.00%	0.00%	6.25%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
10/11/91	0.00%	0.00%	14.58%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	50.00%	0.00%

Figure 4. Dynamic loading for a set of workcentres in output report format

list all the work orders that are scheduled for that date. The portion of the last work order that cannot be finished before midnight of the next date is broken up into two parts: the first part is considered to be the responsibility of the first date, and the remainder is considered as the responsibility of the next date.

Figure 4 shows the dynamic workcentre loading. A number of workcentre codes appear in the figure, from which we shall concentrate on workcentre DRL-W1-HD3. The column for this workcentre is highlighted by a box. The dynamic workcentre loading is the total quantity backlog (column 12 of projected workcentre schedules (Figure 3)) for a particular date (i.e. midnight to next midnight) divided by the capacity of the workcentre. For example, on 10/8/91 the loading of the DRL-W1-HD3 workcentre is $(10000 + ((7.5 \div 12) * 10000)) \div 20000 = 81.25\%$. The second work order begins at 1630 hours on 10/8/91, 7.5 hours out of the total of 12 hours is assigned on 10/8/91, and the remaining 4.5 hours is assigned on 10/9/91. There are no new work orders scheduled for 10/9/91. On 10/9/91 the loading of the same workcentre is $((4.5 \div 12) * 10000) \div 20000 = 18.75\%$. There is no loading on 10/10/91, and on 10/11/91 the loading is $10000 \div 20000 = 50\%$.

Concluding discussion

The main contribution of this project is an efficient implementation of the production order scheduling algorithm, taking into account the production activity control decisions, for a rapidly updating make-to-order production environment by efficient computer memory management and interface to a production database. An attempt is made towards the concept of 'management by the database' instead of 'management of the database', the latter situation referring to the overwhelming problems caused by the unsuccessful use of the database.

Some of the more enhanced features that are being added to this integrated framework are the provisions for splitting work orders. Currently, the work orders cannot be split. Also, queue time estimating methods are being added. Currently, the queue time is input by the user.

An attempt has been made in this paper to describe the main features of a tight integration of production order scheduling with other modules with reference to an actual operational prototype in a product-specific industry. Such JIT based computer-integrated manufacturing concepts can be reused in other industries with a similar type of production environment.

References

1 **Ohno, T** *Toyota Production System: Beyond Large Scale Production*, Productivity Press, NY (1988)

2 **Flapper, S D P, Miltenburg G J and Wijngaard, J** 'Embedding JIT into MRP', *Int. J. Prod. Res.*, Vol 29 No 2 (1991) pp 329–341

3 **Monden, Y** *Toyota Production System*, IIE Press, Norcross, GA (1983)

4 **Vollmann, T E, Berry, W L and Whybark, D C** *Manufacturing Planning and Control Systems*, Irwin, NY (1988)

5 **Sridharan, V and Berry, W L** 'Master production scheduling make-to-stock products: a framework for analysis', *Int. J. Prod. Res.*, Vol 28 No 3 (1990) pp 541–558

6 **Karni, R** 'Capacity requirement planning – a systematization', *Int. J. Prod. Res.*, Vol 20 No 6 (1982) pp 715–739

7 **Orlicky, J A** *Material Requirements Planning*, McGraw Hill, New York (1975)

8 **Biggs, J R** 'Priority rules for shop floor control in a material requirements planning system under various levels of capacity', *Int. J. Prod. Res.*, Vol 23 No 1 (1985) pp 33–46

9 **Hoffman, T R and Scudder, G R** 'Priority scheduling with cost considerations', *Int. J. Prod. Res.*, Vol 21 No 6 (1983) pp 881–889

10 **Anderson, J C and Schroeder, R G** 'Getting results from your MRP system', *Bus. Horizons* (May/June 1984)

11 **White, E M, Anderson, J C, Schroeder, R G and Tupy, S E** 'A study of the MRP implemntation process', *J. Operat. Manage.*, Vol 2 No 3 (1982)

12 **Wight, O W** 'MRP II – Manufacturing resource planning', *Mordern Materials Handling* (September 1979) pp 78–94