Trigger for the next Industrial Revolution Automatic Operations Management

Laxman C Marathe

Trigger for the next Industrial Revolution

Automatic Operations Management

A technology that manages operations at micro level with full future visibility will be the next big change. Production cost will be greatly reduced as cost of coordination is eliminated. Instead of selling products, manufacturers may hire their facilities to several customers on a time-shared basis allowing them to manage what, how much and when to produce what they desire by only paying for the actual time resources were used.

Laxman C. Marathe
Researcher (Factory Physics)

Prelude

Visit any manufacturing setup or factory and you will see something very similar.

The best of ERP system and processes are in place. Even a day's late coming by the most menial worker is accurately recorded using the best of class Human Resource Management module.

Every penny due and recoverable is accounted for by the Finance and Accounts Module. Financial statements are all accurate and made in a jiffy.

A visit to the Stores stuns you. Everything is systematically stored. Inventory and Purchases too are very well managed. There are systems in place everywhere. Name a material and you immediately know how much is in stock, how much is already ordered and so on.

Move to the shop floor. The view is awesome – you see robots and numerically controlled machines that can quickly configure themselves to work on any kind of job. Automatic guided vehicles crisscross the shop floor moving everything around.

Best in class Management Information system permeates the shop floor like a nervous system. All information is captured in real-time and a department is dedicated to manage this real-time "big data" and generate various analytical reports.

There is no dearth of posters and slogans explaining management concepts: Quality First, TQM, 5S, TOC, 2X, SPC, Reengineering, Six sigma, Value Engineering, Kanban, and more - very diligently displayed everywhere indicating emphasis on training and education in techniques that may help streamlining production and improve quality.

You may also see decorated frames at the most prominent places on the shop floor titled "Quality Policy". Your escort will urged you to ask anyone on the shop floor about the company's "Quality Policy" and you are sure to hear a well memorized recitation of the "Quality Policy".

In spite of all the modern technology in place you will still witness that a whole lot of people have flocked together in some conference room overlooking the shop floor, especially if it is the beginning or end of the day. They have all huddled together for what is commonly known as the "Daily Production Meeting [DPM]". The scenes you see and the words you hear may be disturbing at first, however, it is just a matter of getting used to. The more factories you visit, the more familiar you get with these DPMs.

It is this meeting where literally the "who's who" of the factory is present. Every important Manager and Supervisor is in attendance. DPMs are invariably chaired by the BOSS with top sales and marketing personnel as special invitees.

So have they resolved all issues once the meeting is over, would there be no need for such a meeting again? Visit the same shop floor again, same time next day and you will notice an equally boisterous DPM in progress. It is so much a "Daily" occurrence that many just call it a "Production Meeting". The word "Daily" appears both obvious and redundant.

You guessed it right these meetings are very expensive. Just add the man hour cost of such meetings plus the man hour cost for doing preparatory work before and follow-up work after each meeting and it makes a substantial portion of the cost of production; may be as high as 50%.

You are told DPM is the heart and brain of the shop floor: a very true statement. Miss it even for a day and everything literally falls apart. All know that DPMs along with the attendant work it entails is a real cost that is simply unavoidable. Cost of production is actually a sum of cost of "coordinating and managing production" plus the "actual cost incurred in production". One may argue why differentiate between the two when neither is avoidable. However, I wish to claim that it is certainly possible to cut (in fact eliminate) the cost of "coordinating and managing" production. The enabling technology is already in place. You only need to adopt it, provided you know how to recognize it and where to look for it. This booklet is dedicated to this purpose. It is structured logically in four chapters as explained next.

Contents

Daily Production meeting (DPM)

1. Phases of industrial revolution and its effect on DPM

Industrial revolution happened in phases. If you imagine "global industry" as a tree then these stages mark periods of rapid growth and branching. We briefly mention the stages of rapid growth as under.

Powered Mechanization

Machines of sorts were invented and where powered by means way beyond human capability: from steam engines to internal combustion engines and now electric motors. What took ages to be done by hand is done in minutes. Some call it the first phase of rapid industrial growth.

Mass Production

Interchangeable parts, division of labor, assembly lines and the obvious economies of scale achieved thereof. What was once hand crafted for the rich is now available for the masses! Literally, the masses have all become rich. Some say this is the second phase of rapid industrial growth.

An obvious offshoot of mass production was an emphasis on "quality". One can only guarantee interchangeability of parts or the whole, if each is just like the other. Thus adherence to some predetermined standard of "quality" that could be measured became necessary. The word "quality", though popular, appears inappropriate as it has a superlative tint that does not suitably express "adherence to some predetermined measurable standard concept" that it is used to express.

It did not take long for people to realize that "quality" of the final produce can only be guaranteed if and only if the "quality" of the raw material and the "quality" of the value-addition process is duly controlled. Many techniques and standards have helped achieve this objective. One can imagine "quality" to flow through the manufacturing unit – it comes in as "quality" raw material and flows out as "quality" final produce, provided "quality" of value-adding process is controlled.

Automation and networking

Computers and the possibility of networking them was the next big growth trigger. All repetitive and algorithmic activities – anything that could be programmatically controlled or solved using some logic or algorithm no matter how complicated – were and are being automated. Automation guarantees "quality". It speeds up work and is probably cheaper than doing things manually. I used the word **probably** to stress on the fact that automation is perceived to adversely affect employment. What was supposed to help humans by automating mundane mechanical work is considered by many to take away their very source of livelihood.

The internet has blurred the boundaries between everything: be it nations, machines or people. Huge data can now be stored, accessed and updated anytime from anywhere. Connectivity is a given.

The obvious question is what next will drive rapid growth of the "global industry tree". We have many contending candidates as we shall discuss in chapter 3. Most are looking for something NEW. That was how it happened earlier - powerful engines, Mass production techniques and Computers. None of them existed earlier.

The question you may be tempted to ask is how all this is related to DPM, a topic we propose to discuss? Isn't the relationship obvious? The "global Industry tree" is made up of thousands of leaves – each a manufacturing unit in itself that conducts its own DPM ritual that shows no sign of becoming redundant.

Before industrial revolution we had artisans – small groups skilled in making goods ordered for. Even artisans had some sort of DPM but things where simple to manage. A diary, a pencil and a little commonsense was all that was required.

The first growth trigger was powered mechanization requiring reskilling of existing artisans. The DPM ritual mostly remained unaffected.

The second phase of "mass production" changed the nature of DPM. Plans for mass production had to be meticulously drafted and then followed through "daily". DPMs were now mostly entrusted with ensuring that the overall progress do not deviate much from the planned. Still things were manageable as mass production philosophy discouraged variations.

However, the masses were not happy "to have a car of their color choice as long as it was black". Mass production runs contrary to the basic human desire to stand out as someone special and different. Take the case of readymade shirts: one size does not fit all nor does one color or design appeal to all. Though the basic steps in shirt-making remain the same, manufacturers were forced to produce shirts in batches, handled independently for each variation in size, color or design. That impacted the DPM – everything suddenly became very complicated.

All expected the third phase of industrial revolution "computers and networking" to take care of the complexity introduced by variation. Abolish the ritual of DPM once and forever, but that does not seem to be the case. On the contrary, demand for variability is accentuated in this internet savvy world. People now know they have a choice and they are spoilt with several contending options to choose from.

The onus is on the manufacturer to either satiate this hunger for variation else risk obliteration. DPM being the heart of a manufacturing unit is directly overstressed. They are getting increasingly complicated. People in charge of production resent attending one every day but have no option not to. Factory Managers world over are scouting for solutions. The situation is indeed disappointing: the promised benefits fails to materialize, DPMs fail to be trivialized.

2. What happens before, during and after each DPM?

Most preparatory work for DPM involves data collection to cross check if yesterday's target productions were achieved. The management information system comes in handy here as one need not run around the shop floor collecting data. Target productions may be marked on a Gantt chart or stored as a spread sheet listing of department wise jobs for the day.

What is of interest is the deviation. Generally, Operator of each Machine was given a list of jobs or tasks (we shall re-defined these terms later) to be done yesterday. The question is was the list followed? Were all tasks done as expected within the specified time and allowed wastages? If not, then why? Was anything else done? The same exercise must be repeated for all machines. Obviously, each department / section head does it for his machines. So, everyone across the factory is involved.

Department-wise deviation does not present the overall picture, so things must be summarized by someone into meaningful actionable data before the DPM begins.

The meetings generally begin with each department / section head justifying all deviations. The justifications are invariably reasonable and the chair finds it exasperating to note her helplessness in doing anything about it, though things did go haywire.

Sales persons complain – the customer is unhappy as so and so order is delayed and must be expedited, else we be prepared to face the consequences. The Chair shivers at this explicit threat and directs her ire to the last department / section head in the value-adding chain, invariably the dispatch guys, for having delayed things up. The dispatcher has a readymade answer, "didn't received it from the previous step or the quantity was short". The preceding departmental head in the value-adding chain is no newcomer to this blame game anyway. He knows whom to point a finger to. And the game continues…

All is not done yet, marketing representative project a new order load and demand when these new orders will be ready for delivery, obviously with the existing order load considered. The chair directs some able "scheduler", who understands production end-to-end, to figure out when new orders can be delivered. Not an easy task but an answer is still expected almost immediately. So, a quick guestimate is offered which is duly noted as the final delivery date.

The meeting by now is in doldrums, temperature has reached ignition level. The conclusion, however, is simple and always the same. Draw up a new task list department / section wise with urgent tasks listed first. All are warned, like yesterday, that no deviation will be tolerated and things will be reviewed again in the next meeting.

3. Why the need arises to conduct the next DPM tomorrow?

Today's meeting conclusion has an explicit reference to the next meeting as things must be reviewed again anyway. A manufacturing unit is like a living organism that eats "tasks" for food. Every day one has a new set of tasks and new possibilities for failure. So things must be reviewed every day.

What about the list of tasks given each departmental or sectional head. Will they not do the tasks mentioned therein? Why won't' they? Are they not employed for this purpose? Questioning their competence is akin to questioning Factory Management's competence in recruiting able persons. We are actually asking the wrong questions. All shop floor personnel are well trained and have the right equipment and resources to perform all tasks allotted to them. The fault is not in here.

4. So what's going wrong and where?

Will you agree if I say the wrong is all in the list given to each departmental or sectional head? May be some tasks just cannot be done as necessary inputs are not available or there isn't enough time available to do them. May be the allotted task in itself consists of one or more sub-tasks that need to be executed in a specific order.

So, why would one make a list of tasks that is not actionable? Who is to blame – "the Scheduler"? Call her to task!

Let me hasten to again add that the Scheduler too is not to blame. She rather deserves appreciation for at least having created a list of tasks for the day. Know the fault lies elsewhere - a good place to finish Chapter one.

The unsolved problem

1. The planning & scheduling conundrum

Let us begin by an example to show how complex scheduling problem really is. Say you need to bake bread in a community kitchen buying all ingredients yourself. Where do you start? Of course, with the recipe; you must know how to bake bread. Next you have to decide how much bread you need to make? The recipe gives what ingredients you need and may be the time required to bake a 'certain quantity' of bread. If your quantity is different, you must reckon how much to buy of each ingredient. Also some processing times too may change with quantity and one must figure that out as well.

The recipe for baking a pound of bread may look something like this.

Raw Material

1. 3/4 pound refined flour
2. 20 gram crystal sugar
3. 2 gram salt for taste
4. 10 ml vegetable oil
5. 5 gram powdered yeast.
6. 500 ml water

Method

Take 50 ml of water. Add sugar and heat until the sugar dissolves completely.
Wait till it becomes lukewarm. One must be able to insert a finger in it comfortably. Add the powdered yeast and stir. Set aside for at least an hour.

Put the flour and the fermenting yeast solution, salt and oil in the kneading machine and add water to it slowly. Ensure that the dough isn't too tight or too droopy. It must lump together into a ball and must not stick to the kneading machine walls.

Remove the entire dough on a baking tray and shape into a loaf. Ensure the tray is pre-oiled and also brush some oil on the loaf to prevent drying. Cover with a clean wet cloth. Keep the tray to rest in a warm corner of the room for at least two hours. See it rise to twice its original size. Remove the cloth and bake at 200 degree Celsius for 20 minutes. The crust hardens and gets a golden brown hue when fully baked. Remove from oven and brush some oil over the crust for that shiny look. Allow it to cool for an hour before slicing for consumption.

What a tempting recipe. A simple thing like baking bread consists of so many individual tasks that must be done in a particular order to get the end produce "A loaf of bread"

Let us calculate the time needed to make a loaf of bread?

1 hour – Assumption of the time to get all ingredients from the market

1 hour 5 minutes – time to make fermenting yeast solution. You need this before kneading together all ingredients.

Assume 10 minutes to knead a pound of dough. (Add a minute more for each extra pound)
2 hours for the dough to rise
20 minutes for the bread to bake
1 hour for the bread to cool

Now that adds up to five hours thirty five minutes. Even if we remove the one hour cooling time after baking, we still need 4 hours thirty five minutes to make a loaf of bread. Assuming the community kitchen with its kneading machine, its oven and a warm corner are all free only for you to bake a pound of bread then this is the time required. However, the community kitchen is never free, so the actual baking time will be much more. How much more? It is a complex function of the baking load already present and the load you intend to add to the community kitchen.

Imagine how your local baker manages her kitchen. She bakes many a kind of breads and cakes in different quantities and all in the same kitchen. Add to this already complex situation special party orders and one immediately realizes the complexity of the problem. Your local baker too holds her DPM. She too is looking for a solution.

2. Differentiating planning from scheduling – a new interpretation

What is the difference between a plan and a schedule? These two words are always confused. With this example of baking a loaf of bread let me attempt to define each word very clearly. This definition will stand in good stead later and will help recognize the right solution.

The Plan

The recipe to bake bread is not the plan, though many may be tempted to believe so. You must decide what quantity of bread you need to bake. Then break down the baking job into elemental tasks according to the recipe. Calculate amount of input material required. Estimate how much time each task would require: note kneading time is dependent on how much dough is made. Further, for a different community kitchen with a different make and model of the kneading machine this time may change. All this detailing work, when completed, makes “the plan”.

The plan is specific to:

1. What we intend to make “the end product”. Each has its own recipe;
2. The “quantity” we intend to produce and
3. The “place” where we intend to make the end product (the factory).

Change any one of the three and we have a different plan. One can never actually begin work until the “plan” is not ready.

The plan can predict cycle time for creating the end product in a particular factory. However, this cycle time is meaningless as it is calculated assuming infinite resource availability. One cannot even say that this is the minimum time required to create the end product. As we shall learn later some tasks can be overlapped in time to theoretically finish the job earlier.

If you are dealing with just one manufacturing unit "a given factory", the plan changes with the recipe and the quantity. One may thus create several "plans" for different recipes with varying end product quantity as a variable used to calculate all details - quantity of input required, output produced and time required for each task. For example, Plan for white bread X pounds, Plan for brown bread X pounds, Plan for plum cake X pounds. You will learn in Chapter 4 the "Job Study Wizard" (JSW) actually enables one to create such plans where the end product quantity is a variable. Change it and press the "Validate" button to create a specific quantity plan.

Just because one plans to do something meticulously does not mean one has done it. One must schedule a "Plan" for execution.

The schedule

Scheduling is only possible when the "'plan" is ready. The plan calculates the total duration required to execute each task. When one allots specific start and end times – absolute time - for each activity of the "Plan" one gets a schedule. Remember, as things change on the shop floor this schedule must also be redone (rescheduled). Each task will have its start and end time recalculated.

We have to introduce "absolute time" in the Plan to get a schedule. However, absolute time has three parts: past, present and future. So our "planned" activities automatically get classified on the real timeline as:

- Past activities: those already completed
- Presently running activities: Activities still in progress, so we do not know when they would be over. However, we can always anticipate a completion time based on the original estimate of time for this activity in the "plan" and the amount of work already done.
- Future activities – those that are slated to both begin and end sometime in the future.

One factory can have just one schedule for all "plans", each representing a distinct order, being executed concurrently. Meaning "schedule" is always made for the entire factory with all "plans" considered together.

If you agree to this definition of "Plan" and "Schedule" you will immediately realize where the problem is. The past and the present influence the future. The past is done and over with. The future cannot be touched. So as the present changes so does the future. We decide action in the present. So how our present decisions affect the future?

All modern technology at our disposal has got us up-to-date until the present in real-time. But none really allows us to see the future. We need the predictive ability to anticipate impact of our present decisions on the future as fast as the present changes.

3. Why people are not able to crack the scheduling problem?

The first hurdle in solving the scheduling problem starts with the confusion regarding what is the "Plan" and what is the "Schedule". Any solution must address both together.

The bill of material (BOM) concept could have worked well as an integrated planning and scheduling tool. However, BOM emphasized more on the "material" rather than the "activities". Several "activities" may need (or create) the same "material". One can aggregate material together but not the "activities". Each activity must be scheduled independently and that in turn decides what material is required or created when and in what quantity. Activity schedule decide "material" requirement, not the other way round.

Project planning, on the other hand, only maps the activities and is silent on the material aspect. Project plans are graphically represented by joining one activity to another by a line forgetting all about the material flow between the activities.

We need an integrated representation that has activities and the material all in one. As you will read in the fourth chapter the Component Task (CT) diagram is the only perfect "Planning" method as it defines both the material and the activities together unambiguously and in detail. The "Plan" thus represented is ready for automatic scheduling.

The second hurdle is to view the scheduling problem as a text book puzzle. Typical academic approach is to start with framing a "problem statement" and then search for an elegant solution to solve it optimally. The best solution is the winner! That was how we were evaluated in school.

Scheduling is not amenable to such typical academic mindset as the "problem statement" itself keep changing. No point saying let there be 'n' tasks that can be executed on 'm' machines with specific durations and then find an optimal schedule. In real manufacturing units actual durations vary; suddenly a new job comes in or one of the 'm' machines ceases to be available. What good use is this "optimal" schedule then?

The third hurdle is the time required to get an optimum solution. When the "problem statement" is expanded in scope to deal with real manufacturing units having hundreds of jobs, sharing several common resources on the shop floor, it immediately becomes unsolvable as the possibilities are immense. Finding out the best "possible" solution may take ages even using the most powerful computer on earth. Further, this is no one-time exercise. As things change, everything needs to be rescheduled.

What is our objective? To solve some fictitious "scheduling problem" or help the "Factory manager" by making her life easier and cutting the cost of coordinating production. Why not ask the factory Manager instead what she needs.

4. What the factory manager needs?

Having attended DPM myself, I can say for sure having an "optimum schedule" is not what is required. The Factory Managers needs to know the final impact of everything. Expressed in simple terms:

1. The earliest possible completion time for each order in hand, given the situation as it is NOW: with the constraints, the inefficiencies, breakdowns, reported delays all considered. Obviously, this prediction must be revised as fast as the present conditions change.
2. If a new order comes in when this new order will be completed earliest given the existing load.

So requirement is for an ability to predict the future within the confines of the factory for all orders in hand. And this prediction must be redone as fast as current circumstances change. Do you feel this is an impossible thing to achieve? Let me assure you it is not so. One can do so easily. The solution to achieve this objective is already available. One only has to start using it. Needless to say, manufacturing units that employ this solution will leave their competitors far behind in the race.

However, before we actually concentrate on the solution and where to get it from, we must review many catchy words and concepts doing rounds in the industry and academia. See them in the penetrating light of our new understanding, in order to separate the chaff from the grain. Doing so is necessary as we must carry forward what is good and dump the rest.

Current tools, techniques and technology

We follow an alphabetic order to review the current tools, techniques and technology available. So this chapter is like a bibliography of terms. The idea is to separate the grain of truth from the chaff of nonsense. We must always look at the relevance of each from the point of view of making DPM redundant. The prefix tilde and hash tag has a meaning as explained at the end of this chapter.

~3D Printing

3D printing is also called additive printing as one prints thin layer upon layer that eventually takes the shape of a 3D object. This printing is done using some specially formulated material whose flow can be controlled while printing but eventually hardens to protect the form.

It is a wonderful technology and in my opinion a great help in the designing stages as intricate 3D forms can be produced quickly and with great accuracy. It is a tool for the imaginative mind to sculpt something new. No more painstaking shaping, carving or chiseling just "3D print" them. Further, once the form is finalized duplicating it too is proverbially just one button click away.

Many people think 3D printing is science-fiction coming true. Press a button on a 3D printer named "Strawberry ice cream" and lo you get a huge bowl filled with the stuff ready to eat. Next press another adjacent button named "Stretch Limousine" and wow a huge automobile rolls out of the 3D printer door. Hop in it and go for a long drive. *Do not forget to pick the ice-cream bowl before embarking on the long drive.*

Most real things we use are complex, made up of thousands of different substances, metals, chemical compounds with varied shapes and sizes all organized into one complex whole. Our science fiction 3D printers will have to work at the sub-atomic particle level to create, atom-by-atom, literally anything. The next step is to have a 3D printer clone itself – a reproductive 3D printer. Then we become "God" with the ability to create anything. Why just DPM, the entire industry tree becomes redundant in such a case.

5S Principle

5S is the name of a workplace organization method that uses a list of five Japanese words: seiri, seiton, seiso, seiketsu, and shitsuke. In English they mean "sort", "set in order", "shine", "standardize", and "sustain". A sixth word "Safety" is also added in by some now. Obviously, 5S is important but that does not make DPM redundant.

#Advance Planning and Scheduling (APS)

Going by text book definition APS is what we need to really make DPM redundant. It is the way forward but as you will soon realize, many APS systems stumbled when they came to

"Production scheduling". The same hurdles to cracking the scheduling problem beset any APS solutions created until now. The only way out is to have a truly automatic scheduling engine to drive the APS. You will see such a scheduling engine is now readily available.

~Automation

We already mentioned how "computers" had a great impact. Anything that could be programmatically controlled or solved using some logic or algorithm no matter how complicated – was and is done. Machines and computers are linked through transducers, sensors and actuators. The Computer aligns, does the set-up and actually works the machine to finish the job, all without human intervention. Automation is so advanced that it can even fly planes to a pre-determined destination and make them land. However, humans must still decide what job to do or where to go!

For automation to make DPM redundant one must be able to program the "ability to decide what should be the next task" on each machine on the shop floor? This again brings us back full circle to "scheduling" and all the hurdles involved in cracking this problem. The relevant question to ask is "can scheduling a complex factory be automated". If yes, then that is the real game changer. The purpose of this booklet is to impress upon you that it can be done, in fact, it has been done.

#Bill of materials (BOM)

What goes in as sub-parts in the end product? The sub-parts in turn may require many more sub-parts to make and so on until we arrive at the raw material needed to be procured. So BOM is actually a part product plan with material and quantity defined.

As already mentioned the BOM concept came very close to the actual solution we shall discuss next. However, the activities that put these sub-parts together were forgotten. The CT diagram representation, we discuss in chapter 4, overcomes this deficiency.

#Bottlenecks & Constraints

Visit any manufacturing setup and you are sure to find some machines / equipment used almost round the clock, whereas some others remain virtually idle.

Take the example of our community kitchen. Add a refrigerator to the community kitchen in addition to the existing kneading machine and the oven. If on a cold rainy day all only want to bake something then the refrigerator will remain unused, whereas on a hot summer day the refrigerator may become the bottleneck.

If one takes a decision to sell off the refrigerator in winter, one is sure to repent when summer comes. Likewise, if in the heat of peak summer one hurries buying another refrigerator then both will lie idle through the winter, whereas the oven will still remain a bottleneck. So what? Buy another oven to take care of winter months. This is what the theory of constraints actually recommends as we shall discuss later.

You will realize that bottlenecks or constraints are a function of the order mix – baking orders vis-à-vis ice-cream orders? Change the order mix and the bottlenecks change.

It may be a good thing to do some line balancing in an assembly-line kind of situation where all orders pass through each station of the line. Adding a bottleneck machine / equipment may yield permanent results. However, the “assembly line approach” does not work in a job shop kind of environment with ever changing order mix each requiring a different routing. Doing so may be simply a waste of money.

Identifying bottlenecks is easy, any machine that has a pile-up of work-in-progress before it. However, the pile-up is an effect of having erred in accepting orders. We need a solution that forewarns us about a potential bottleneck. Rather work around the bottleneck and anticipate when the order would be ready for delivery.

In a job shop one may find several bottlenecks. One cannot pin-point one of them as “the” bottleneck. What is important is to know is how bottlenecks impact orders. In simple words, when each order will be ready, given whatever the bottlenecks or constraints that may exist. To answer this question we again need the ability to schedule a complex factory. That brings us back to cracking the scheduling problem.

<u>#Capacity and efficiency</u>

Like bottleneck, capacity and efficiency too are throwbacks to the “assembly line approach” of thinking. The entire line’s capacity will equal the bottleneck station capacity: the weakest link in the chain.

Frankly speaking “scheduling” is not a problem in an assembly line kind of an industry where large numbers of the same product are made. There is really nothing to schedule. The whole factory is like a “pipe”; water goes in from one end and exits the other end. May be the end product is an assembly of several sub-part each with its own sub-assembly line that confluence into one final line. No big deal, each feeding sub-part line too has a fixed feeding rate which can be calculated as a function of end product producing line’s production rate. It is a one-time job. But do such manufacturing units exist anymore?

In a job shop like situation, where each order is different and follows its own peculiar routing there is no point talking about capacity. Instead we must estimation how much time each activity of each order will take to complete on each machine / workstation. This is indeed a detailed exercise that must be done while “Planning” an order. As you will understand in chapter 4 the CT diagram representation helps automate this complex task.

<u>#Critical chain management</u>

Critical path is something calculated assuming infinite availability of resources. One never has infinite resources in real life. So, the actual time to complete a project, especially in a multi-project environment, where different project activities contend to share common limited resources, is more than what the critical path envisages.

What is this additional time required would have been clear if the Scheduling problem was cracked? Then one is in a position to directly answer when each project will be completed. That is the real solution. However, if scheduling a complex setup is not possible, then how do manufacturers commit a delivery date? They propose a very safe date that is several times more than the "real time" required for executing the order. The next obvious question would be, "why customers agree to such an extended delivery schedule"? Frankly, they have no option. All manufacturers are in the same boat. No one has a means to predict the real delivery date. "Scheduling problem" is not solved for anyone.

Mr. Eliyahu M. Goldratt, the originator of Theory of Constraint philosophy, was smart to realize that in order to play safe, manufacturing Managers world over commit a disproportionably long time to delivery then is really required. Meaning they have huge time "buffers" to play with. He worked the reverse way, by saying first identify these excess time buffers you have luckily got. Monitor how much you have consumed of these time buffers for each order. Expedite those orders that have consumed the maximum buffer already to ensure all project finish as committed.

However, this approach will be rendered useless when the scheduling problem is cracked. Buffers will shrink dramatically as manufactures will compete on delivery time, which becomes the new "unique selling point".

#Drag and Drop feature

Have you come across a scheduling package that does not allow any drag & drop feature, probably not? You are made to believe it is an important feature that is a 'must'. Follow the schedule as it is and observe the mess it creates.

Look a little deeper and you will realize this is no feature at all, rather an inconvenience imposed upon you. It is the weakness of the scheduling software to create a schedule that is not "actionable". One is expected to correct it painstakingly and thus the drag & drop facility, provided under the guise of a glorious feature.

Drag & drop is a manual intervention requiring substantial time to be finalized by just one person. So rescheduling cannot be done as fast as things change on the shop floor. All decision making gets concentrated in hands of this one person who must painstakingly release the final schedule for all to follow. How often can this exercise be repeated?

As you will soon realize non-existence of a drag & drop facility is the litmus test for a truly automatic scheduling capability. Search for a scheduling system that does not provide any drag & drop feature.

#Enterprise resource planning (ERP)

There was a time when everyone wanted an ERP installed in a mad rush to automate, thinking this is the ultimate solution to all problems and the result will be a dramatic increase in profits and manufacturing efficiency. As years pass, factory managers have started realizing that nothing of that sort has happened. Many are even wondering if the investment in an ERP system had any real positive impact at all.

Has the ritual of DPM discontinued after having installed an ERP system? No! Many ERPs claim to provide a Scheduling module. However, the schedule created is neither detailed nor actionable. One must do a detailed schedule manually by taking help of a spread sheet, or a Gantt chart board.

ERP vendor rather stress on automating peripheral activities like purchase, finance, human resource management, customer relationship management first. Doing so has limited benefits as the heart (day-to-day scheduling) that drives the entire organization is still mostly a manual activity.

#Finite capacity scheduling

Ask a counter question, "What does infinite capacity planning mean"? In fact "scheduling" is required primarily because capacity is finite and the same finite capacity must be used to accomplish several concurrent contending orders. This is a basic functionality expected not something that requires a special mention.

#Flexible manufacturing systems (FMS)

It is the ability of the machines and the material movement systems to be re-configured in order to create a different end product or movement path. Configuration information may flow from a central server to all machines and movement vehicles so that they can ready themselves to work on something different.

However, the FMS does not decide what should be produced NOW and next on each machine, which in turn also decides what material must be moved around. Someone still has to order a change that the FMS may faithfully follow. The DPM continue to retain its original importance.

#Internet of things (IoT)

Computers and networking is so inexpensive and commonplace that many have started imagining a world where everything is connected to everything else. They wonder what such ubiquitous interconnectivity and processing power might throw up. Many imagine this nebulous, ill-defined concept as IoT. Each is free to interpret what it means and how things would become different. I would only call it as an IoT balloon of imagination. Just a pin prick and it pops.

Here are the pins to prick the IoT balloon:
Machines may talk to each other but who decides what they must do?
What are the boundaries of this interconnectivity? Won't it be a security threat if everything is widely interconnected?

Finally how does IoT impacts DPM, which we have set out to make irrelevant, is not really clear.

#Inventory Control

Inventory is of three types:

Raw material – something you purchase from market and if still in pristine shape may still have some "sale value".

Work-in-progress – something you create en route while producing an end product. This generally has no "sale value".

End product – something you intend to sell. If sold will command full market value else is a total waste.

It is a common mindset of factory managers to see that everybody and everything is always working. They cannot stomach a person or a machine standing idle. It appears an obvious waste of precious resources. So doing something, even if it is of no immediate use, is the main reason for unwanted inventory.

Look at the problem closely and you will realize it has its root in factory managers' inability to foresee the future. If the manager could view a Gantt chart confirming all orders will be delivered as committed and still huge idle periods exists on certain resources, then she may rather shut idle resources temporarily and reassign the manpower to do something else, instead of forcing them to produce what is not required immediately. However, in order to predict the future one must first crack the scheduling problem. An immediate benefit of automatic scheduling will be just-in-time-inventory-creation.

Another reason for creation of unwanted inventory is erroneous demand forecasting. One forecasts a heavy demand for item 'A' with little requirement for item 'B'. Obviously, one produces more of item 'A' and little of item 'B'. If the forecast goes wrong: item 'B' is more in demand and not item 'A'. Then the surplus inventory of item 'A' remains unsold whereas one loses business as supply of item 'B' is insufficient to meet demand.

Even in such situations automated scheduling can help, as it allows one to produce in small batches without making production activity in any way more complex. Smaller batch size is less risky, as it allows frequent corrections to suit changing market demand.

#JIT / Lean Manufacturing / Kanban

These are management principles derived from Toyota Production System. Just in Time (JIT) is defined as a management system of producing only what is required, in the exact quantity required, exactly when it is required and delivered exactly where it is required. The question one now needs to answer is who decides when what is required and in what quantity. In an assembly line kind of a scenario this was a relatively easy thing to achieve by what is called Kanbans. They are visual signals telling the preceding process that something needs to be produced as it is no more seen on a shelf or a flag indicates it is required. These visual signals flow from the end product until the initial raw material procurement stage.

Obviously, if one produces only when required (indicated by a Kanban signal) and in the quantity needed (again passed through the signal in some way) unwanted wasteful production is curbed. Idle workers or machines are no more considered as wasting time rather as waiting for a Kanban signal.

Benefits of having a lean manufacturing system are obvious: just-in-time production and elimination of unwanted inventory. However doing so in a job shop like situation with hundreds of varied orders, each with its own peculiar routing, would be an immensely complex and an impossible to implement task.

Instead, if the scheduling problem is cracked, and it is possible to reschedule a complex factory as fast as things change, then the same "scheduling engine" can issue Kanbans to initiate all production actions just-in-time. As you will see in the next chapter this is exactly what happens.

#Line balancing

Again a throwback to pure assembly line kind of thinking where it works and mostly done as a one-time exercise. We have already discussed that it is impossible to balance a complex job shop. Bottlenecks depend on the current order mix.

#Load Leveling

We discussed "Finite capacity scheduling" as a basic scheduling feature. However, most scheduling systems available do not perform finite capacity scheduling. Meaning they overload a workstation. Like in an eight hours shift one is assigned 200 hours of work. What will one do? Do what one can in 8 hours and leave the rest for tomorrow. That is exactly what load leveling is all about. However, it takes manual intervention to decide what to do today and what to postpone for tomorrow.

#Make span

If one looks at scheduling as a text book problem with a pre-defined set of jobs then one of the objective could be to reduce the make span of all jobs. That is how academicians address the scheduling problem.

Let us start with a schedule which has all jobs having the minimum possible make span. As we start executing individual activities, some get over faster others are delayed. Eventually the schedule is out of synchronization with the real situation on the shop floor. All assumptions about minimizing make span are rendered meaningless.

#Manufacturing planning and control

The system concerned with planning and controlling all aspects of manufacturing, including managing materials, scheduling machines and people, and coordinating suppliers and key customers.

Obviously, this is what we finally need to achieve by introducing an integrated system that is built around an automatic scheduling engine core. However, for this to happen, the scheduling problem must be solved first. Read chapter 4 to know how it can be done.

Material handling

Bulk of material movement on the shop floor is of work-in-progress (WiP). Output from one workstation must be moved over to the next when required. In a complex job shop, every

WiP produced may not be immediately required and thus must be stored until needed and moved to the right place when required. Contrast this with an assembly line where the material flow itself controls production.

In order to coordinate material movement in a job shop the next activity on each work station must be decided in order to initiate real-time movement of material. An automatic scheduling engine accomplishes this goal as explained in chapter 4.

#MRP & MRP II

Tough the acronyms appear similar they mean very different things.

The first term stands for material requirement planning. If one needs an end product in a particular quantity then using the bill of material (BOM), one can decide what are the sub-parts and subsequent sub-parts required until we reach the raw material stage.

As already discussed, BOM concept is silent about activities. To perform activities we need resources. Resources are limited and shared. So, the concept of MRP II came in. Here the 'M' stands for manufacturing, whereas 'R' is the resource that we intend to plan. In short we need to create a schedule of activities in accordance with the availability of resources. But then it means we must crack the scheduling problem. However, that is not what MRP II does. It simply works out a timetable of when what would be required by working backwards in the BOM hierarchy that is now infused with activity times estimate data. The activity time estimate is done independently for each order / project without consideration to other orders: as if one has infinite resources availability.

#PERT and CPM

PERT (Program Evaluation Review Technique) and CPM (Critical Path Method) are project management techniques, which have been created out of the need of Western industrial and military establishments to plan, schedule and control complex projects. The greatest drawback being it is silent on material flow and is limited to just one project control at a time. Scheduling is still required when multiple projects vie for shared and limited resources.

You will soon learn that the CT diagram is the only way out, as it takes care of activities and material in one single representation. What is defined as a CT diagram can be directly scheduled because it is a complete "Plan" definition.

Poka-yoke

In simple English it means "inadvertent error prevention". Mark the word 'inadvertent'. Though poka-yoke has no relation to "scheduling" it is still important to us as DPM also discusses "quality" issues. What is the point in producing something in time with defects? Further, reworks and repairs will affect the delivery dates adversely.

Poka-Yoke is a mindset: think what could go wrong inadvertently and try to design or arrange things such that the wrong is never allowed to happen.

Predetermined motion & time study

These are work measurement *systems* based on the analysis of work into basic human movements, classified according to the nature of each movement and the conditions under which it is made.

Such studies will help improve efficiencies of the production process and reduce activity times as well as cost.

Predictive Maintenance

A stitch in time saves nine. If one can predict while the equipment / machine is still in-use what maintenance work is needed and do it, then one prevents a potential break down.

Preventive maintenance increases accuracy of predicted delivery date as the factory becomes reliable.

Quality Control / Assurance and Standard compliance

To survive in business the quality of end product must be guaranteed. ‘ Quality Assurance’ is a Strategy of Prevention, whereas ‘Quality Control’ is a Strategy for detection of deviations and one must follow a standard practice to ensure it happens so every day.

DPM also discusses quality issues as rework and repair work disturbs a schedule negatively.

Reengineering, Value Engineering

Professor Michael Hammer of MIT claimed: most of the work being done does not add any value for customers, and this work should be removed, not accelerated through automation. It called for a relook at all the business process we undertake rather than assuming them as a given and trying to do them faster using technology.

Value engineering (VE) is a systematic method to improve the "*value*" of goods or products and services by examination of its functions. *Value* is defined as the ratio of “function” to “cost”. *Value* can therefore be increased by either improving the “function” or reducing the “cost”.

Value engineering deals with the design of the end product, whereas reengineering concentrates on the process of creating one. Both influence a “Product Plan”. If we eliminate unwanted product features or activities then we certainly hasten creation time as well as reduce costs.

#Theory Of Constraints

Theory of Constraint assumes there is always a constraint that restricts and controls the throughput of the factory. Initially the word constraint only meant “bottleneck equipment” but was later expanded to mean even “people” and “policies”. The idea is to identify and maximize use of the constraint resource or to elevate it so that throughput increases adding to profit.

The problem with TOC is that only one constraint can be targeted at a time. Recollect the Kitchen example mentioned earlier, where the job mix decides what will become a constraint. In real complex job shops, the job mix and constraints keeps shifting. One cannot force a resource to be the "constraint" just because one desires to implement TOC.

<u>Conclusion</u>

We discussed 28 tools, techniques and technologies. You will find some 19 have a hash tags prefix. They will all cease to matter when the scheduling problem will be solved. Out of the remaining nine two have a tilde prefix as they are not connected with scheduling. The remaining seven would still remain relevant.

If automatic scheduling were possible!

Rendered irrelevant	Will remain relevant
#Advance Planning and Scheduling (APS)	5S
#Bill of materials (BOM)	Material handling
#Bottlenecks & Constraints	Poka-yoke
#Capacity and efficiency	Predetermined motion time study
#Critical chain management	Predictive Maintenance
#Drag and Drop feature	Quality Control / assurance and Standard compliance
#Enterprise resource planning (ERP)	Reengineering, Value Engineering
#Finite capacity scheduling	
#Flexible manufacturing systems (FMS)	
#Internet of things (IoT)	
#Inventory Control	
#JIT / Lean Manufacturing / Kanban	
#Line balancing	
#Load Leveling	
#Make span	
#Manufacturing planning and control	
#MRP & MRP II	
#PERT and CPM	
#Theory Of Constraints	

The solution

1. The CT diagram

We have already mentioned that the Bill of Material concept is silent on the activities. Similarly, the project graph, where one activity is joined to another by a line, does not give any idea of the material flow. In fact these deficient representations forced a distinction between a "production job" and a "project". We now describe the CT diagram that handles both the activities as well as the material in one representation.

Any job or project, no matter how complicated, can always be expressed in terms of three basic entities: Components, Tasks and Workstations. We defined each terms below.

Component

Component is an "entity" that is needed to accomplish a task or that which is created as an output of a task. Component need not always be something material, it can also be a concept or a design.

Components have an "extent", i.e. the number of identical, indistinguishable items / repetitions it represents. Minimum "extent" of a component must be '1' and is always a whole number. Component "extent" is not its physical quantity. We shall explain this term later.

Components have a conception time and a birth time. When the first instance / repetition of a component come into existence, we mark it as the conception time, whereas when the entire (last) extent is created we call it the birth time.

The end product we sell, or services we render to our customers too are components. So are the raw materials we purchase, and the work-in-progress we create en route. Thus, component is a generic term used to represent anything that is consumed or created when any activity, hereinafter defined as "task", is performed.

Task

We define "Task" as an elemental quantum of work that can be executed on one workstation in one-go. Tasks consume (uses or requires) one or more input components and produces one or more output components. Tasks are value-adding activities. They transform or in some way modify the input component.

"Polishing something" is a task as it accepts "something unpolished" and produces the same thing but now "polished". Likewise, "Assembling something" task takes as input several distinct components and produced one "something assembled". Both tasks "Polishing Something" and "Assembling Something" need a workstation where it can be performed.

Workstation

Workstation is some equipment, persons or a combination of both. Workstation has a position on the shop floor. A workstation need not be always present on the shop floor. It can come into existence on demand and then be dismantled or removed once the task is executed. However, we always know what workstations exist or can be created on demand in a factory.

A workstation can execute either one or several tasks concurrently. A task engages a workstation, or part of it, for a specified period. The same task when executed on another workstation may require a different time or incur a different wastage.

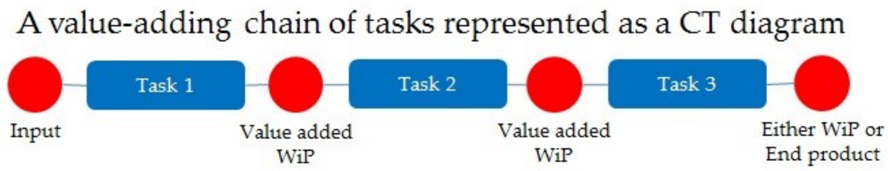
Job or project

With CT diagram representation there is no need to make a distinction between a job and a project. When the output is something intangible, like a design or a service or something big like a building or a bridge one tends to call it a project, whereas when some material end product is created we tend to call it a job. The concept of "component" unifies both tangible and intangible outputs as one. Henceforth, we just call it a "job".

It is wrong to say we do a "job". We perform all tasks that make the job. We accomplish a "job" as a result of doing all tasks defined for the job.

The CT diagram

Here the acronym CT stands for "Component" and "Task". Actually even the workstation is defined in a CT diagram though not explicitly seen. A component is depicted as a circle whereas a task by a rectangle. The task links itself to various workstations of the factory on which it can be executed. One may imagine them as listed below the task and thus hidden from view.

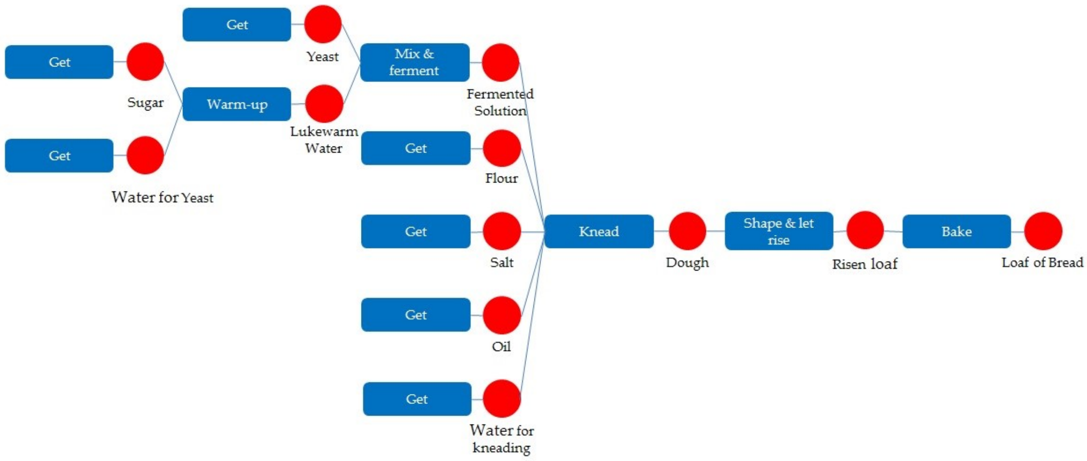


In a CT Diagram, we always start from the left and move towards right as we keep adding value to the input components. The following rules apply for all CT diagrams.

1. We cannot have a stand-alone component or a stand-alone task they must be always linked.
2. CT diagram always starts with a task and ends with a component. The first task is always a "get something task" whereas the final component is invariably the "end product". So, procurement too is part of the CT diagram definition.
3. A task can consume one or more input components and can produce one or more output components. However, each task must produce at least one component.
4. Component to component and task to task joining is not allowed.

5. A component can be consumed by just one task and must be an output of just one task.
6. Circular joining is not allowed.

We now represent our bread baking job as a CT diagram.



You can immediately notice how detailed the representation is. We have twelve tasks and twelve components. Each task can be executed on a workstation in one-go.

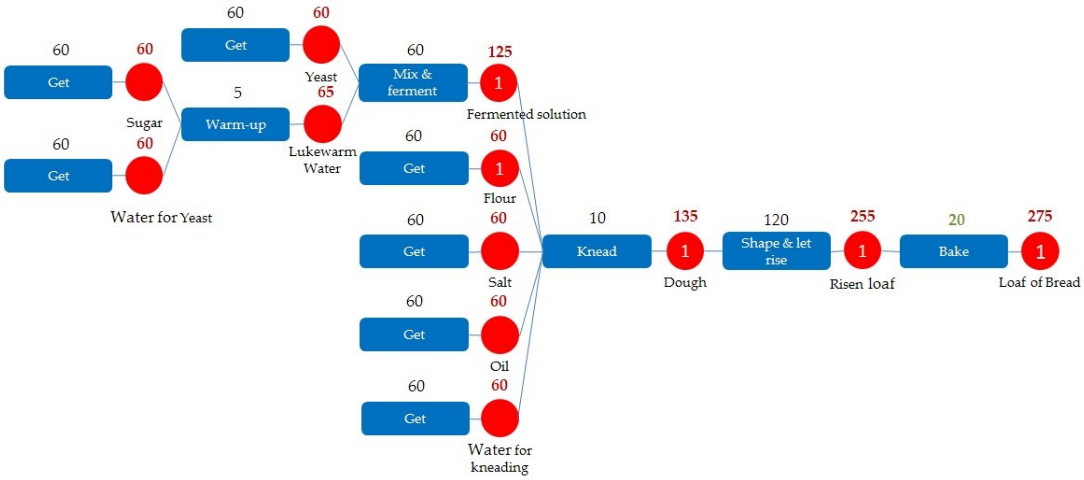
Out of the twelve tasks seven are "get" input tasks. So, procurement too is defined in a CT diagram. These tasks execute on a workstation "Stores & Purchase". Stores & Purchase workstation may handle multiple tasks concurrently. Task "warm-up" requires a small stove. Task "Knead" a kneading machine whereas task "Bake" an oven. On the other hand task "Mix & ferment" and "shape & let rise" are purely manual tasks.

The mapping of tasks to workstations of a factory is a one-time job. One does so while defining what is called the "factory database". It is a master database that contains a lot more information about the factory. One can define attributes for tasks, components and workstations as well as for the job in the factory database. One can write formulae to calculate the duration, wastage and cost of executing each task on each workstation based on the extent of the output component as well any of the attributes already defined.

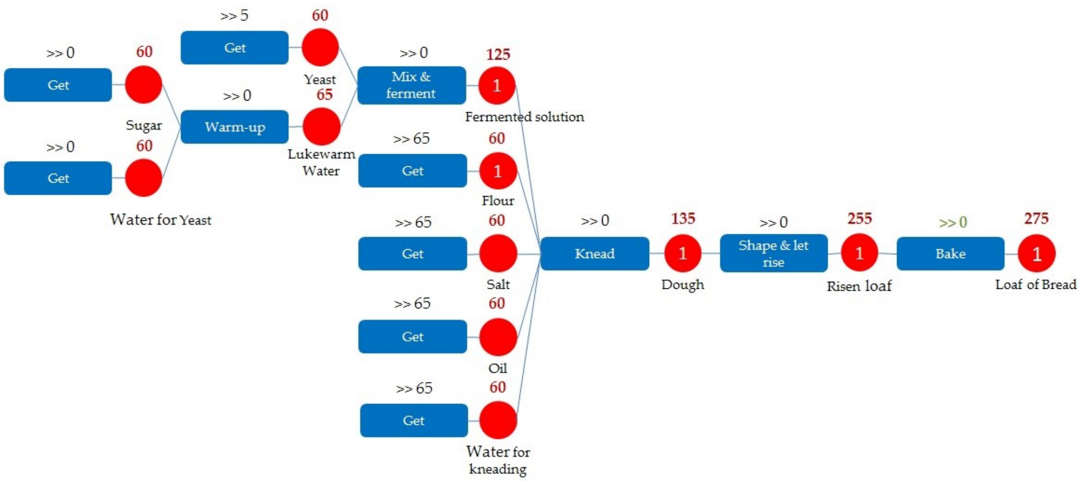
We now explain the concept of component extent. Luckily for our "loaf of bread" job we have not assumed any wastage so the extent of each component will be the same. If one wish to make 'n' loafs of bread then we can work backwards calculating the extent of input components required. They will be 'n' all across as no wastage is considered. We can always get the actual quantity of raw material by multiplying the quantity required for '1' loaf of bread by the value of 'n'. The table next gives quantity of raw material required for making 10 loafs of bread.

Raw Material	How much to make 1 loaf of bread as defined in the recipe	How much to make 10 loafs of bread
Flour	0.75 pounds	7.5 pounds
Sugar	20 gram	200 gram
Salt	2 gram	20 gram
Oil	10 ml	100 ml
Yeast	5 gram	50 gram
Water for Yeast	50 ml	500 ml (1/2 liter)
Water for kneading	450 ml	4500 ml (4.5 liter)

As we know how much time each task requires, we can use this information to calculate the time-to-birth for each component in the CT diagram as under.



The figure above each task is the task duration in minutes, whereas the figure above each component is the time-to-birth in minutes. We can now proceed to calculate the feeding buffer available for each input raw material. We call it the "intrinsic leeway" available as it is an intrinsic property of the very recipe for creation of the end produce. The raw material is not required until the intrinsic leeway is over. On the contrary, if procured earlier may lie unused.



The value above each task prefixed with a double arrow gives the intrinsic leeway available in minutes. You can very easily guess that Water for kneading, Oil, Salt and flour is not required until 65 minutes from the moment one decides to make a loaf of bread, whereas, yeast can come in after 5 minutes. Instinctively, we know that our bread making must start with heating water added with sugar to make a lukewarm solution in which to culture the yeast. The CT diagram representation confirms this is the first step indeed.

Once the intrinsic leeway is known it is possible to calculate the future burden on each task of the job. If more than one task of a job vies for execution on the same workstation then the one with a higher future burden is the one to be executed first to ensure the job is finished earlier.

Let us understand the concept of future burden by an example. Say, we need to make a jam sandwich but have a toaster that can toast just one bread slice at a time. We toast that slice first on which we are to apply the butter and jam rather than the slice we use to close our sandwich with. We instinctively start applying butter and jam on the first toasted slice while simultaneously inserting the second slice in the toaster without realizing that the task "toast" first slice has a higher future burden, as some more value-addition is still required before joining it with the closing slice. Our sandwich job has two "toast" tasks.

The factory database has formulae defined to calculate duration, wastage and cost for each task and each task-to-workstation combination. So, we can perform all these detailed calculations automatically. We get the cost and time-to-birth for each component including the end produce. So we estimate the cost and cycle time for the job. Additionally, a host of other aspects of the job as well as for each task, component and workstation too can be calculated. One gets to know the extent for each component, and thus the actual quantities of raw material required, each task's duration on each workstation it can be executed on, task to workstation wise wastages, the cost of each component and the time-to-birth as well as the leeway. One can thus have a complete "plan" when the job is represented as a CT diagram.

As you will see in the solution, an entire module called the "job study wizard" (JSW) is dedicated to define jobs as a CT diagram. One can define a generic recipe with some extent and then convert it to a specific "plan" for any end product quantity by just mentioning a desired quantity and clicking the "Validate" button. All calculations are performed automatically according to the new end product extent.

When you use the JSW you will learn how easy it is to define a new job. One can define "parts" in a job and then copy them over as basic building blocks to create new jobs. One can copy creating chains up to a particular component and then replicate them as many times as desired to make a complex job. JSW automatically validates the job and ensures that what you "Plan" is actually executable on the shop floor defined by the factory database.

One can create templates for all kinds of jobs undertaken. When an order actually materializes, one just needs to mention the required end product quantity and re-validate a pre-defined job. Jobs can be saved as a file and directly scheduled for execution.

JSW thus allows one to schedule already defined and valid jobs. It also has a wonderful feature to simulate what-if scenarios by requesting a snap-shot of the scheduling engine's current status. One must actually experience and feel the kind of features available in the JSW.

2. How one schedules once orders are defined as CT diagrams

The scheduling engine is a black box for the user and works in the background. The JSW is an interface one uses to schedule a job. One can have hundreds of JSWs in a factory each working independent of the other.

In the scheduling engine all jobs from all JSWs are treated alike. Scheduling engine is only concerned with the overall task load. It is the task that must be scheduled. Jobs' getting done is a result of having completed all tasks that make the job.

Scheduling engine's overall logic can be expressed in one simple statement – If a task **can be done** and it **should be done** then it will be done. The "can be done" part encompasses many aspects – availability of input components, at least in quantities enough to begin the task, availability of an active workstation and material movement confirmation. Whereas, "should be done" is a choice of user to delay task execution until the leeway expires. If both the conditions are satisfied then the task is allotted for execution. If more than one task vie for execution on the same workstation, then job priority and the future burden of the task is used to select the first allotted task. Job priority is something for the user to decide and change anytime.

This is a very simplistic view of the scheduling engine's logic. In addition to the above, many more execution decisions are taken. We list them below.

Execution decision	What it means	Scheduling engine usage
WIP control	Attempt to minimize work-in-progress (WIP) from being created too much in advance and thus remain unused.	If WIP is not on the critical chain and has enough time left to be produced and used then its creation is deferred thereby minimizing WIP build-up on the shop floor.
Control of task execution order	Honoring user's desires to change task execution order at run-time. Using this feature is not recommended.	Tasks are allotted first by job priority and then by the future burden on the task within a job. However, user may change this natural order of execution at run time.
Workstation choice	If one has a choice of workstations to perform a task then which one to choose?	Scheduler tries to honors user preference with switchover savings, if any, considered. In case the first preferred workstation is unavailable it tries to allot the task on the second preferred one and so on.
Locking Option	Ensuring a particular task is only executed within a user specified period.	Always tries to execute the said task within the specified period, as far as possible.

Execution decision	What it means	Scheduling engine usage
Auto-breaking option	Breaking up a task to run concurrently on more than one workstation in order to reduce task execution time.	If the task is on the critical chain or its execution cannot be deferred any further scheduling engine will try to optimize and select the most appropriate breaking option possible.
Spanning Option	Stop and resume task execution after a holiday, recess period. Commonly referred to as a non-scheduling time zone (NSTZ) in the system.	Scheduler wisely decides to span or not to span depending on the current situation.
MCI option	It may not be necessary to wait to start the next value-adding task that uses or consumes what is produced by the current task, until the current task is not over. One can overlap in time both tasks in order to expedite the job. We can say the preceding task gives a mid-course intimation (MCI) to the next task to begin.	Scheduler tries to begin the next value adding task even before the earlier one, feeding into the next, is not yet over. Time to initiate the next task can either be user decided or left to the scheduling engine to figure out.
Interleaving option	User may want some jobs to be executed only when there is free time available. Contrast this with auto-breaking option where the objective was to expedite.	Scheduler ensures the tasks of such jobs are executed whenever there is nothing urgent to be done.
MCF Option	Especially in long running tasks interim milestone reached feedback may be necessary to re-adjust expected task completion time. We call it a mid-course feedback (MCF).	MCF is used constructively to adjudge the expected completion time for long running tasks.
NSTZ cut-in option	NSTZ is an acronym for non-scheduling time zones. Periods when the scheduling engine will not schedule (allot) a fresh task. However, an already running task can either by design (or because it is delayed) cut-into an impending NSTZ. System supports five levels of NSTZ with varying importance and user can define how much a particular task can actually cut into each of them.	Scheduling engine takes appropriate decision to cut into NSTZ whenever necessary. Working during NSTZ is an additional cost and this cost is also calculated.
Task line-up	It is possible for user to specify that in a job if some task A is executed first on a particular workstation then preferably task B too should be the next one followed by task C and so on. Valid reason for doing so could be substantial saving in cost and time. We call it task cascading. This again could be a preferential cascading or a forced cascading when user insists that the scheduling engine waits a pre-determined period for the next cascaded task to mature for execution.	Scheduling engine honors these cascading requirements as well.

3. Four components of the solution

The entire system comprises of just four modules. At the heart is the scheduling engine with a master copy of the factory database. One factory can have just one scheduling engine and one factory database.

Job Study wizard (JSW)

One can have several JSWs in a factory – may be one for each sales person. As already discussed JSW helps one to define a new job and also schedule them. Each JSW also acts like an independent simulation module allowing one to conduct what-if analysis based on the most current snap shot of the job load requested from the scheduling engine.

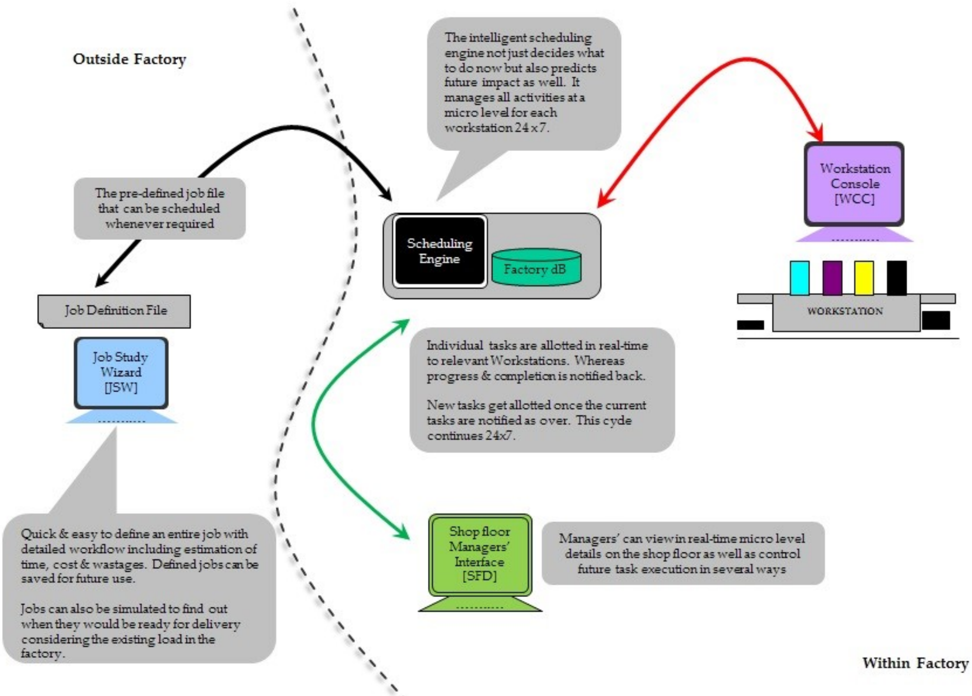
Workstation Console (WCC)

When the scheduling engine schedules a task, the task information (like a Kanban signal) flows electronically to the relevant WCC on the shop floor. WCC is the Operator's interface with the scheduling engine. The Operator is told what task to execute now in real time, as well as notify its progress and completion back to the scheduling engine. One could have several WCCs on the shop floor, each dedicated to service a single or a group of workstations as desired by the factory manager.

Shop floor display (SFD)

This is the Manager's console. The manager can monitor jobs' progress in real-time and also control how future task execution may be done. One can have several SFD consoles each in real-time synchronization with the scheduling engine.

How four modules of the system work together



4. Where you can find such a system to experiment with it?

A windows version of the system named "Manuka" is available for free download from any of the links below. Do try installing it on your PC. All help documentation too is embedded in the installation.

http://www.mediafire.com/download/q5q80a1o1ogp7qf/ManukaLiveSetup.rar
http://www.mediafire.com/download/tphuv0275qhi3ao/ManukaLiveSetup.zip

Uncompressing the downloaded file will yield

1. A folder named "ManukaLive" that has all the installable files;
2. A read me text file named "ReadMeFirst" giving details of how to install the system and;
3. The actual setup program called "ManukaLiveSetup.exe".

Click on the ManukaLiveSetup to install the software.

A folder named "Help" in the installed directory has complete details about the system as a whole and for each of the four modules. Strongly urge you to study this material to acquaint yourself with the working of the system.

Help folder contents:

1. About Henry Namgyel's Carpentry Shop.pdf [pages 10]
2. Basic Concepts.pdf [pages 6]
3. CCF_Manager Quick Reference Guide.pdf -- [pages 1]
4. JSW Quick Reference Guide.pdf [Pages 14]
5. SFD Quick Reference Guide.pdf [Pages 9]
6. nWCC Quick Reference Guide.pdf [Pages 12]
7. Simulator Quick Reference Guide.pdf [Pages 9]
8. How To Uninstall Manuka Live.txt [1/2 Page]
9. Reporting Tool Quick Reference Guide.pdf [pages 8]
10. How2CreateFactoryDB.pdf [pages 81]

An updated copy of this book is available from the link:
http://www.mediafire.com/download/p4k4293k029ckwt/Trigger4thIndustrialRevolution%282%29.pdf

About the author

The author has about 30 years of industrial experience when he had ample opportunity to understand how daily production meetings are conducted and how an absence of a scheduling system forces concerned people to take day-to-day decisions based on experience and sheer guesswork.

The concept of the CT diagram took shape in the author's mind way back in 1999. Since then the author is behind the concept – developing it, refining it – until it has reached a working system stage that you can now download and use.

The author is a double graduate: in Physics and then in electronics engineering. He post-graduated in Industrial Management.

You can reach him:

By post: Laxman Chandrakant Marathe
"Manisha Bunglow", Plot No 49, C.S.No. 251/1+2,
Abhaynagar, Sangli – 416416, Maharashtra, India.

By email: laxman.marathe@gmail.com

By phone: (+91) 9011098634 or 9910128634