

Brought to you by www.ScienceofBusiness.com, the site with all your Goldratt Theory of Constraints information – books, self learning materials, [Dr Lisa](#) products, speaker information

Science of Business Goldratt Implementation Group US

Knowledge Jolt with Jack

Jack Vinson writes about knowledge management, personal effectiveness, theory of constraints and more. Knowledge Jolt provides consulting services to businesses and individuals with a focus on establishing effective work practices.

TOC ICO, Simplified Drum-Buffer-Rope

The second (and final) session I attended Tuesday was another by Eli Schragenheim, this time describing how Simplified Drum Buffer Rope (S-DBR) works, how it was developed and how it relates to traditional DBR. This was particularly interesting, as Eli Schragenheim gets the credit for conceptualizing and developing S-DBR. If you'd rather read it from the source, check out this article from 2000, [Simplified Drum-Buffer-Rope](#), by Eli Schragenheim and Bill Dettmer. If you want my detailed take on it, keep reading.

The key new knowledge for me is that traditional DBR gets the sequence wrong. This discussion and that of Eli Goldratt earlier in the conference makes it clear that it is the Rope first, then the Buffer, and the Drum comes into play only in special cases. The other key is that the full solution needs to synchronize Sales and Operations.

Drum-Buffer-Rope (DBR) is a mechanism for running a production environment with the focus on exploiting the constraint of the operation. In practice this involves setting a detailed work schedule for the constraint (drum), buffering the constraint so that it is never starved (buffer), and setting a release mechanism to ensure that work gets released into the system at the right time (rope). Schragenheim also stated that there was a hidden assumption in the development of DBR: that sales and operations operate in silos - sales provides the orders with little to no input from operations. In the current thinking, this assumption has to be broken. The true constraint of the system is the market, and operations must subordinate to that fact.

What happens if we explore the impact of DBR with the knowledge that production is no longer the constraint of the business? Schragenheim went through some of the main parts of DBR as they are currently practiced, and suggested that most of them are no longer necessary. First, the finite schedule for the constraint: having a detailed schedule of work for the constraint ensures that the constraint is always active (exploit). It also creates a policy of "adherence to the schedule." While these are good things for the constraint in the plant, they make the plant inflexible to new demands from the market: a customer needs an order sooner, a stock order becomes critical, etc.

Second, traditional DBR requires three kinds of buffers: a shipping buffer to protect the order due-date, the constraint buffer to protect the schedule at the constraint, and assembly buffers to ensure post-constraint assemblies get all their non-constraint parts quickly. It turns out that most DBR doesn't use the assembly buffers, and Schragenheim suggests that they really operated as additional protection of the shipping buffer. The primary issue Schragenheim talked about was that DBR provides no direct mechanism for prioritizing the signals from all of these buffers. Primarily, the problem is: do I ship or do I keep the constraint busy?

Third, DBR is too complex! There are a number of issues that no DBR software can handle totally: dependent setups and other technical limitations require part sequencing; operations like ovens that work

Science of Business ♦ Goldratt Implementation Group US
303.909.3343 phone ♦ 303.362.7353 fax ♦ www.ScienceofBusiness.com ♦ www.Viable-Vision.com
Info@ScienceofBusiness.com ♦ 3845 Forest Street, Denver CO 80207

Brought to you by www.ScienceofBusiness.com, the site with all your Goldratt Theory of Constraints information – books, self learning materials, [Dr Lisa](#) products, speaker information

Science of Business Goldratt Implementation Group US

several orders (or partial orders) together have complex rules for scheduling; if the constraint consists of a number of similar-but-not-identical machines, the scheduling problem is complicated; situations where orders go through the constraint several times (or through two constraints).

Finally, in practice there is rescheduling due to the reasons above or to our friend Murphy. With the finite schedule at the constraint, this can create changes all over, namely that the promised due-dates could slip. This is exactly the opposite of exploiting the market constraint. Since no DBR software handles all of this complexity, there are always side systems that attempt to make-up for the lack in the DBR software, leading to complication at the least and lack of confidence in DBR at the worst.

Yikes. This is too much bad news. Isn't DBR supposed to be a good thing? Schragenheim argues that while the spirit of DBR is simple, the implementation and all the special cases has made it more complex than it needs to be. In classic TOC thinking, this is a conflict cloud for a "successful solution:" do something simple to get good enough execution vs. do something sophisticated to get the utmost from execution. Break the conflict by checking how much the results are impacted by the simple solution vs. the sophisticated solution. There are three areas where a simple solution might create problems:

1. We might starve the constraint in production. If the true constraint of the business is the market, this means that the capacity in production has to be higher than that of the market demand. If so, then we should expect the plant to be operating below full capacity, meaning that the constraint is not 100% activated all the time.
2. The order in which work arrives at the constraint causes too much down time, wasting the capacity of the production constraint. This is relevant when dependent setups have a major impact, which is fairly rare. In these cases, we might need a more DBR-like solution.
3. The constraint works on things that are not truly needed. This should not happen if the work released to the shop is related to market demand (actual orders), and the work is released according to the release signals set by the Rope.

Basic Design of S-DBR: What are the core pieces of DBR that we'd like to keep? Schragenheim talked about three primary aspects:

1. be able to know whether the delivery dates are safe (subordination to the market).
2. choke release of material onto the floor to ensure the constraint gets material on time without creating piles of work-in-process (Rope).
3. smooth the load on the whole shop.

In order to satisfy #1, Sales gets a tool that gives them a quick answer to the question, "what delivery date will this order have?" The standard answer will be the standard lead time that the company quotes. However, the tool calculates this by placing the order on the constraint at its next available time and then adding 1/2 the shipping buffer. If this is beyond the standard lead time, this new (longer) time will be quoted, as it is more likely to be correct. If it is shorter, then the standard lead time is given, and the order is given a longer order buffer to ensure that the order will come across the constraint at about the right place. (Yes, each order gets its own "order buffer" that helps control execution.)

In order to satisfy #2 and #3, there have to be simple rules for the shop floor. Every order has its order buffer that are set on submission of the order. Work is released into the system a buffer time before it is due, just like the Rope in DBR has always worked. And priorities are set by the color of the order buffer.

Brought to you by www.ScienceofBusiness.com, the site with all your Goldratt Theory of Constraints information – books, self learning materials, [Dr Lisa](#) products, speaker information

Science of Business Goldratt Implementation Group US

Work centers will have this information on a daily basis to prioritize their activities, and there will only be one set of signals, rather than signals from Assembly, Constraint and Shipping buffers.

But, but, but this seems far too simple. Won't this system create far too much work-in-process? Isn't there a danger of overloading (or starving) the constraint without a detailed schedule of work? What if there are emergency orders or other changes to the current orders? (Schrageheim had a nice list of what-if's.) The first two issues are addressed above.

How to handle emergency orders? If you are in an environment where you expect emergency orders, then a portion of the planned load should be set aside for these kinds of orders. Then, when Sales checks "can we deliver early," the tool should still be able to provide good estimates of early due dates.

If order priorities change after they have been entered into the system, the mechanism is to change the order due date. The shop-floor priority mechanism will take care of the rest. For example, if an order is needed a week earlier, the due date will be moved up, and the buffer calculation will show a new priority. No need for sophisticated rescheduling of the detailed constraint schedule. Nice.

There are a couple watchout areas in this design. If the facility runs both make-to-order and make-to-stock, the MTS orders shift priorities based on the consumption from the stock buffers. If there is not protection in the system for these changes, then we can impact due-date performance of make-to-order work. The other place to be careful is with dependent setups - setups that take significantly longer/shorter depending on the sequence of work that runs across through the work center. The first thing to check is whether the setups can be improved so that operations doesn't have so much trouble shifting from part-to-part. Another option is to identify a preferred sequence of work and let the foreman make decisions on the fly as to the sequence of orders with the understanding that the order buffer is long enough to allow for these adjustments. Schrageheim suggested that we should shift back to traditional DBR only as a last resort, when these measures aren't good enough to deal with significant wasted time at the constraint.

So, in conclusion, I may not understand all of the ramifications of S-DBR, but I certainly have a better idea than I did a week ago. As I stated above, the essence in Operations is that there is one buffer that sets all the priority and the release schedule. Sales uses the same information to schedule orders through the system. Simple and powerful.

[Permalink](#) | written on [9 Nov 2006](#) at 5:57 PM | [Comments \(0\)](#) | Subjects: [event+report](#), [theory+of+constraints](#), [DBR](#), [EliSchrageheim](#), [tocico](#)