

Is There a Poor "Herbie" anywhere in the Oracle SCM Area?

Member Question:

From the "Theory of Constraints" there is poor Herbie, the infamous bottleneck in the supply chain pipeline. For any of you who use the Oracle Supply Chain 11.5.9 (or above) platform, is there any built-in "Herbie" that you've noticed, or has it been custom code wrapped around vanilla, that effected you most?

New York City, U.S.

My Response:

After reading your question, I could not help but think of my Dad at Christmas time when he painstakingly had to test each bulb to find the one that did not work and replace it before the entire string would light up. (Note: for those of you born within the last 25 to 30 years, this will make absolutely no sense beyond the analogy itself.)

Introduced by Eliyahu M. Glodratt in his 1984 book titled "The Goal," the inherent flaw in the Theory of Constraints (TOC) philosophy is similar to that of the old "chain" of Christmas lights in which one bulb could bring down the entire string. In short, and in accordance with the TOC's five steps of focus that are supposedly designed to stimulate ongoing process improvement, the theory confines or limits you to work within the framework of a flawed architecture.

This means that after identifying the constraint, the champion is then supposed to find a way to exploit, subordinate and elevate it. The TOC philosophy then makes the assumption that when one constraint has been removed another constraint will likely arise thereby requiring a fifth step, in which ongoing vigilance is recommended. It is reminiscent of the days of dBASE II coding.

With the latter, when you repaired one bug in a line of code, there was a strong possibility that another line which had previously worked without problems would fail. What this meant is that you were in perpetual debugging mode where the system (and its ongoing challenges) became the focal point rather than the business itself. Or as one senior purchasing official from South Dakota lamented, "are we in the software business or the procurement business?" after having to dedicate 2 full time resources to making their PeopleSoft application work.

I have commonly referred to this as being a linear or sequential architecture in which an equation-based model has been used as a framework for application development.

What is interesting is that the basic principles of TOC, which are centered on a Convergency-Consistency-Respect axiom, actually touch on the periphery of the foundational elements of an agent-based model in which the synchronization of seemingly disparate elements are a reflection of the real-world processes that defines the organization. The reverse if you will of the TOC's five step "program."

Using the Christmas lights analogy once again, with the strings that are sold today one bulb's failure (think bottleneck) does not bring the entire string down. Nor does it require a perpetual state of vigilance in terms of having to seek or "identify" an obfuscated malfunctioning bulb.

While I am not necessarily sold completely on the veracity of Object Oriented Programming (which is a discussion for another day), its very modularity reflects the independently connected light string that is represented in the analogy.

Given the above, under which scenario would you prefer to operate; a) in an environment where the architecture is sequentially connected and therefore requires you to monitor all lights in the "chain," or b) an environment which is based upon a synchronized (or Metaprise) architecture where a bottleneck in one area of the operation does not have a degradative effect on the entire network.

Now while the above examples are indicative of an all or nothing proposition (re the lights either work or don't work), in the real-world supply chain practice there are varying degrees of "performance greys" as I call them. However, this hopefully illustrates my point relative to your question.

What is also worth noting is that the problems with sequential architectures are well known to vendors such as Oracle, SAP and Ariba to name just a few. With 85% of all initiatives failing to achieve the expected results, it would be hard to miss.

However, and similar to TOC, these vendors are confining their client base to work within the limited framework of a flawed architecture as demonstrated by the introduction of Service Oriented Architectures (SOA). Oracle is of course championing Project Fusion, while SAP's response to bridging the synchronized connectivity gap is called Safe Passage. At best, these are little more than static "band-aide" solutions in which an attempt is being made to make current sequential-centric ERP platforms interact more effectively at an operational level. And like Object Oriented Programming is a topic for discussion in and of itself, so to is the subject of Service Oriented Architectures.

As a point of reference I am pleased to provide you with the links to a number of articles I have written that will help to shed some additional light on your question surrounding TOC.

Dangerous Supply Chain Myths (Part 6)

<http://procureinsights.wordpress.com/2007/06/28/dangerous-supply-chain-myths-part-6/>

Double Marginalization and the Decentralized Supply Chain

<http://procureinsights.wordpress.com/2007/08/09/double-marginalization-and-the-decentralized-supply-chain/>

Optimization Modeling and the Modern Supply Chain (A PI Q and A)

<http://procureinsights.wordpress.com/2008/03/18/optimization-modeling-and-the-modern-supply-chain-a-pi-q-and-a/>

Popular SpeedQuest™ Service Now Available

<http://procureinsights.wordpress.com/2008/03/19/popular-speedquest%e2%84%a2-service-now-available-to-pi-blog-readers/>

Author's LinkedIn Profile: <http://www.linkedin.com/in/jwhansen>