**Computing Requirements
For Dynamically
Optimized Systems**

In the world today we continually hear of the automation of everything: walking machines, space probes, voice recognition, and driverless motorcycles. One thing each of these endeavors has in common is the "operating environment" of the device is unpredictable. In order for the device to function, the device must automatically adapt to conditions, as they currently exist.

Our industry, the automation of distribution or fulfillment centers, was one of the first to "automate". Some of us can remember the days when our inventory (information) was kept (stored) on cards and a "new inventory control system" was a new set of drawers where we keep the cards, or a newly hired inventory clerk.

Being one of the first to automate has its benefits and drawbacks. The benefits are the experience it gives us, the drawbacks are also the experience it gives us. The drawback of our experience is that we set in our minds ways of thinking that may be no longer beneficial. Trying to avoid too much controversy in this statement, we will provide a single example. How do we handle a "lost" item in an inventory system? Inventory systems have both financial and fulfillment implications that are at "odds" with each other. Count the times you have seen "lost" items "planned" to be "sold". Wouldn't it be nice if we could just cut an invoice stating: "Dear Customer, we lost the item you ordered so please accept this IOU until it is found. We know it is here somewhere. Thanks for your patience, Sincerely, Customer Service." Lost items have absolutely no value to the distribution process. Likewise, found items should have immediate value to the distribution system. However, trying to create such a "distribution" inventory control system today creates such a pushback from finance due to "their inherited experience" of how automation should work. Finance inherently operates on data, and their view is that reality needs to conform to the data, and that we can "plan" from that data. Conversely, distribution operates on reality and their view is that the data is to conform to reality, and while something may be "planned" the execution of that plan is subject to change.

By now you should be asking: What has all of this to do with "Computing Requirements For Dynamically Optimized Systems"? Dynamically optimized systems use a technique called "concurrent planning and execution". Concurrent planning means that the "plan" is rolling forth as it is executed. The automated devices identified in the first paragraph of this paper are examples of automation where "concurrent planning and execution" are required. The example of the driverless motorcycle demonstrates this condition. The motorcycle takes a different path each time it passes the same area. It is "trying" to go on the same path, but unforeseen conditions cause it to tilt slightly, a puff of wind, a bump, and then the correction. But the correction requires a change in the path.

These are exactly the same conditions that exist in a distribution center. We may have a great plan but "hiccups" force the path to change. The programmer of the motorcycle could easily devise a program (plan) that "knew" all of the conditions that would be encountered, and then executed the "plan". If the motorcycle hit a rock that was not part of the plan, the programmer could say: "It is not the fault of the program at all, the rock was not identified. Once we put the rock in the plan the motorcycle will work perfectly". This is exactly how most distribution systems are operated today. The plan works perfectly and efficiently as long as it is executed properly.

Dynamic optimization or concurrent planning and execution is the alternative to static planning and perfect execution. This however comes at a

cost. It takes more computing power to dynamically plan and execute than to batch plan and execute.

At VAS we use a "scalable" computing platform that allows us to add computing power as necessary to insure that we have adequate resources to make timely adjustments to the unfolding plan. We have a great example of the "scalability" of the computing platform that we use. In a very large project (300,000 single units a day delivered in orders ranging in size from 10-50 units) was designed in the mid 90's using "state of the art" Intel 486-100's. To implement the project a server cluster of 30 CPU's was used. In the ensuing years, the CPU's were replaced with faster machines. The computing hardware was reduced to 9 servers and not a single program had to be modified to make the change. Why 9 machines rather than 3 or 4, after all the CPUs are over ten times faster? Because of the speed improvements of the disk drives have only increased three fold.

The scalable platform VAS uses is called MandateIP®. It runs on Intel CPUs under Linux (or Windows). MandateIP® is a derivative of a product called Mandate that has it roots back into the mid 1970's where engineers at VAS installed their first distribution systems. MandateIP® and its predecessors have 100's of man-years of development. MandateIP® is stable providing reliable delivery of billions of dollars of good annually.

Could VAS competitors provide software that concurrently plans and executes? Probably so, but as of now, no one to our knowledge have platforms that support such a structure. Why? They have come from a different set of experiences where the "view of the world" is that reality is to conform to the data. This is a typical programmer's view. A view where the when "plan (program) is correct", it will operate perfectly if the execution is done properly. A

great real example of this mind set, is an operations person was questioning a very intelligent programmer concerning the cumbersome method the programmer had implemented to resolve an exception. When questioned, the programmer responded that it was done intentionally, the "real problem" was that a mistake had been made by a worker to create the situation and if it were too easy to correct, it would only encourage more mistakes. Usually a few well chosen questions will enable you to determine if a system provider understands the necessity of dynamic optimization in fulfillment operations.

At VAS our view and experiences are much different. We believe that reality is just "what is" and that the data should model reality and be adjusted to continually match what is known. This enables us to deliver systems where we concurrently plan and execute to achieve the goal at hand.

What are the "Computing Requirements For Dynamically Optimized Systems"? – A scalable platform with timely support of the decisions necessary to reach the desired objectives. This is MandateIP®!