In 1995, the American Society for Quality began providing professional certification in quality management—the Certified Quality Manager (CQM)—through closed-book testing on a pre-determined body of knowledge. That body of knowledge now includes the Theory of Constraints (TOC).

To aid prospective quality professionals in preparing for the CQM examination, Quality America, Inc. (Phoenix, AZ) prepared a comprehensive book on the CQM body of knowledge called The Complete Guide to the CQM, written by Thomas Pyzdek and originally published in 1996.

The following paper is a chapter in the updated 2000 edition of The CQM Guide. This paper is copyrighted material. ©Quality America, 2000. Reprinted with permission.

Constraint Management

by H. William Dettmer

INTRODUCTION

As we enter the twenty-first century, "quality" has come to encompass much more than just product or process conformance. More and more, quality is in the eye of the beholder: the customer. And with the widespread acceptance of higher product and process quality standards as a way of life among commercial companies, not-for-profit organizations, and government agencies, the playing field tends to level out to some degree. For the customer, this means that it can be more difficult to differentiate between the product quality offered by competitors. When this happens, it's only natural that a customer's attention should shift to another characteristic of the interaction with a supplier: quality of service. ("Sure, they deliver a quality product, but their service leaves a lot to be desired.")

In many cases, the difference between winning and losing a customer's business (or continued loyalty) resides in the quality of the interaction between customer and supplier. In other words, service can be as important an element in customer satisfaction as product quality. Once managers venture beyond product and process conformance (or quality), they move up from process improvement to system-level improvement.

More than ever before, quality managers are finding themselves in need of system-level tools in order to sustain the relevance in business success they've fought so hard to achieve. One such system-level tool is constraint management. Constraint management doesn't assume that product and process quality have already been taken care of. Rather, it acknowledges that quality is but one important element in the business equation. Constraint management seeks to help managers at all levels of an organization maintain proper focus on the factors that are most critical to overall success: *system constraints*. In some systems, these might be quality-related. In other systems, they may extend well beyond the traditional territory of quality.

Have you ever heard the saying, "You can't put ten pounds of sand in a five-pound bag?" How about, "I'm trying to juggle too many priorities at once." If you've heard (or said) either of these things, you have some sense of the stress that constraints impose on everyone. In many cases, the challenge appears to be not enough time. But the characteristics of time are immutable. Excluding considerations of the theory of relativity, time passes at a constant, familiar, well-defined rate—though sometimes it seems to positively fly, while other times it "drags" interminably! So the causes of stress must really be something else. This "something else" is often some kind of constraint.

Sometimes our constraints aren't merely *physical* (e.g., lack of space, not enough resources, etc.). In many cases they're *policies:* the laws, regulations, rules, or procedures that determine what we can or can't do. Even the way we think can be a constraint to ourselves and our organizations. Who hasn't heard it said, "That's the way we do things around here"? Or, alternatively, "That's *not* the way we do things around here." What you're hearing is the verbalization of a policy, possibly unwritten, but accepted as traditional practice nonetheless. When a policy of any kind inhibits what we need (or want) to accomplish, it, too, constitutes a constraint.

The practice of constraint management tacitly recognizes that constraints limit what we can do in any circumstance, and it provides the vehicle to understand why this happens and what can be done about the constraints we face.

Constraint management is an outgrowth of the Theory of Constraints (TOC), a set of principles and concepts introduced by Eliyahu M. Goldratt, an Israeli physicist, in the 1980s in a book entitled *The Goal* (Goldratt, 1986). These principles and concepts are a blend of both existing and new ideas. The new ideas build upon older ones to produce a robust, holistic approach to understanding and managing complex systems. To extend the theoretical principles and concepts into application, Goldratt developed three classes of tools, which will be described in more detail later. For now, the important point to remember is that TOC, and constraint management as a whole, constitutes a *systems management* methodology.

THE SYSTEMS APPROACH

What do we mean by *systems management?* Throughout the twentieth century, management thought was largely activity-oriented. In the early 1900s, Frederick Taylor's scientific management (Taylor, 1947) focused on dividing and subdividing work into discrete tasks or activities that could be closely monitored, measured, and "tweaked" to produce the most efficient performance from each activity. By the second half of the century, the focus had enlarged somewhat to encompass managing processes composed of several activities. At some level, these processes could become quite large and complex, such as a production process, a purchasing process, or a marketing and sales process.

One way of dealing with complexity is to compartmentalize it—to cut it up into "manageable bites." Organizations typically do that by creating functional departments. Each department is responsible for some function that constitutes a part of the whole system. One could even say that these "parts" are actually individual processes. This is an orderly way to come to grips with the issue of complexity.

Throughout the 1980s and early 1990s, the meteoric rise of the quality movement reinforced the idea that success lay in continuous refinement of processes. The ultimate objective became "six sigma," a level of defect-free performance unheard of twenty years before. Unquestionably, both commercial and non-commercial organizations needed this focus. Poor product quality (which is usually a result of faulty process quality) can bring down an organization faster than just about anything else. But many companies, despite herculean efforts and the expenditure of significant amounts of money, were disappointed to find that their payback wasn't what they expected it to be. The idea that if you "build a better mousetrap, the world will beat a path to your door" worked exceptionally well for the companies whose overriding constraint had been product quality. Yet for other companies, the strategy seemed to be somewhat underwhelming.

Despite the admonition to "consider your internal customer," many departments still behave as if they're in a "silo" by themselves. They pay lip service to the idea, but for a variety of reasons they don't practice it very well. Their focus remains inward, on individual measures of performance and efficiency. Most efforts are spent improving the links of the supply chain, with little effort devoted to the linkages, or interfaces between links, and the operation of the chain as a whole.

Systems Thinking

What these companies failed to appreciate was that a higher level of thinking was needed: *systems thinking.* Once the quality of individual processes is put reasonably well into line, other factors emerge to warrant attention. Consider the analogy of a football team.

Major professional sports spend a lot of time and money on process improvement, even though they probably don't look at it that way. A team owner can spend millions on a contract for a star quarterback. By applying natural talent, they expect to "improve the passing process." But in many cases, the touchdowns don't appear, despite the huge sums spent on star quarterbacks. At some point in the "process failure mode effects analysis," the coaches discover it's impossible for this highly-valued quarterback to complete passes from flat on his back. They find that the offensive line needs shoring up. Or a good blocking back is needed, or a better game plan, or any number of other factors.

The point is that any organization, like a football team, succeeds or fails as a complete system, not as a collection of isolated, independent parts or processes. In the same way that a motion picture clip tells us much more about a situation than an instantaneous snapshot, systems thinking gives us a clearer picture of the whole organizational dynamic. In *The Fifth Discipline* (Senge, 1990), Peter Senge proposes that the only sustainable competitive advantage comes from transforming a company into a "learning organization." The keys to doing this, Senge maintains, are five basic disciplines that every organization striving for success must master: systems thinking, personal mastery, mental models, building a shared vision, and team learning. Guess which one he considers the most important. Though he numbers it fifth, he lists it first, and he titled his book after it.

System Optimization Versus Process Improvement

If one "thinks system," the question inevitably arises: What do we do with process improvement? Do we ignore it now that we're thinking at a higher level? No, process improvement is *still* important. It constitutes the building blocks upon which system performance is based. But like the football team alluded to above, once you have a "star performer" at every position, you have a challenge of a different sort: coordinating and synchronizing the efforts of every component in the system to produce the best *system* result. In other words, once the ducks are in line, the task is to make them march in step together. We refer to this as *system optimization.*

How important is it to optimize the system, rather than its component parts? Deming himself answered that question in *The New Economics for Industry, Government, Education* (Deming, 1993, pp. 53, 100). He observed:

> *Optimization is the process of orchestrating the efforts of all components toward achievement of the stated aim. Optimization is management's job. Everybody wins with optimization.*
>
> *Anything less than optimization of the system will bring eventual loss to every component in the system. Any group should have as its aim optimization of the larger system that the group operates in.*

> *The obligation of any component is to contribute its best to the system, not to maximize its own production, profit, or sales, nor any other competitive measure. Some components may operate at a loss themselves in order to optimize the whole system, including the components that take a loss.*

This is a powerful indictment of the way most companies have been doing business since Frederick Taylor's time, not excluding the "quality enlightenment" era of the 1980s and 1990s. In essence, Deming said that maximizing local efficiencies everywhere in a system is not necessarily a good thing to do.

Systems As Chains

To express the concept of system constraints more simply, Goldratt has equated systems to chains (Goldratt, 1990, p.53):

> *We are dealing here with "chains" of actions. What determines the performance of a chain? The strength of the chain is determined by the strength of its weakest link. How many weakest links exist in a chain? As long as statistical fluctuations prevent the links from being totally identical, there is only one weakest link in a chain.*

Goldratt goes on to suggest that there are as many constraints in a system as there are truly independent chains. Realistically, in most systems there aren't very many truly independent chains. The dictionary (Barnes and Noble, 1989) defines *system* as:

> *an assemblage or combination of things or parts forming a complex or unitary whole; the structure or organization, society, business...*

Thomas H. Athey defines a system as any set of components which could be seen as working together for the overall objective of the whole (Athey, 1982, p.12). The underlying theme in these definitions is an interrelatedness or interdependency. By definition, then, a "system" can't have too many truly independent chains. So if there aren't too many independent chains in a particular system—whether a manufacturing, service, or government system—at any given time, only a very few variables truly determine the performance of the system.

This idea has profound implications for managers. If only a very few variables determine system performance, the complexity of managers' jobs can be dramatically reduced. Look at it in terms of the Pareto rule, which suggests that only 20 percent of a system accounts for 80 percent of the problems within it. If this is a valid conclusion, managers should be able to concentrate most of their attention on that critical 20 percent. Goldratt's concept of chains and "weakest links" takes the Pareto concept a step farther: the weakest link accounts for 99 percent of the success or failure of a system to progress toward its goal (Goldratt, 1990, p. 53).

BASIC CONSTRAINT MANAGEMENT PRINCIPLES AND CONCEPTS

Constraint management exhibits the theoretical foundation that Deming considered so important to effective management action (Deming, 1986). Theories can be either *descriptive* or *prescriptive*. A descriptive theory generally tells only why things are the way they are. It doesn't provide any guidance for what to *do* about the information it provides. An example of descriptive theory might be Newton's laws of gravitation, or Einstein's theories of relativity. Prescriptive theory describes, too, but it also guides through prescribed actions. Most management theories are prescriptive. The Deming philosophy prescribes through its fourteen points (Deming, 1986). Ken Blanchard's *One-Minute Manager* (Blanchard, 1988) and Peter Senge's *The Fifth Discipline* (Senge, 1990) explain prescriptive theories in detail. Constraint management is prescriptive as well. It provides a common definition of a constraint, four basic underlying assumptions, and five focusing steps to guide management action.

Definition of a Constraint

Simply put, a constraint is *anything that limits a system* (company or agency) *in reaching its goal* (Goldratt, 1990, pp. 56-57). This is a very broad definition, because it encompasses a wide variety of possible constraining elements. Constraints could be physical (equipment, facilities, material, people), or they could be policies (laws, regulations, or the way we choose to do business—or choose *not* to do business). Frequently, policies cause physical constraints to appear.

Types of Constraints

Identifying and breaking constraints becomes a little easier if there is an orderly way of classifying them. From the preceding discussion, we know that system constraints can be considered either physical or policy. Within those two broad categories, there are seven basic types (Schragenheim and Dettmer. 2000, ch. 4):

Market. Not enough demand for a product or service.
Resource. Not enough people, equipment, or facilities to satisfy the demand for products or services.
Material. Inability to obtain required materials in the quantity or quality needed to satisfy the demand for products or services.

Supplier/vendor. Unreliability (inconsistency) of a supplier or vendor, or excessive lead time in responding to orders.
Financial. Insufficient cash flow to sustain an operation. For example, a company that can't produce more until payment has been received for work previously completed, because they might need that revenue to purchase materials for a firm order that's waiting.
Knowledge/Competence. Knowledge: Information or knowledge to improve business performance is not resident within the system or organization. Competence: People don't have the skills (or skill levels) necessary to perform at higher levels required to remain competitive.
Policy. Any law, regulation, rule, or business practice that inhibits progress toward the system's goal.

*NOTE: In most cases, a policy is most likely behind a constraint from any of the first six categories. For this reason, the Theory of Constraints assigns a very high importance to policy analysis, which will be discussed in more detail under **The Logical Thinking Process,** below.*

Not all of these types apply to all systems. Material and supplier/vendor constraints might not apply to service organizations. Market constraints are generally not relevant in not-for-profit systems, such as government agencies. But resource, financial, knowledge/competence, and policy constraints can potentially affect all types of organizations.

Four Underlying Assumptions

Constraint management is based on four assumptions about how systems function. (Schragenheim and Dettmer, 2000, ch.2) These assumptions are:

1. Every system has a goal and a finite set of necessary conditions that must be satisfied to achieve that goal. Effective effort to improve system performance is not possible without a clear understanding and consensus about what the goal and necessary conditions are.

2. The sum of a system's local optima does not equal the global system optimum. In other words, the most effective system does not come from maximizing the efficiency of each system component individually, without regard to its interaction with other components.

3. Very few variables—maybe only one—limit the performance of a system at any one time. This is equivalent to the "weakest link" concept discussed earlier.

4. All systems are subject to logical cause-and-effect. There are natural and logical consequences to any action, decision, or event. For those events that

have already occurred, these consequences can be visually mapped to aid in situation or problem analysis. For those decisions that have yet to occur, or which are contemplated, the outcomes of these actions, decisions or events can be logically projected into the future and visually mapped as well.

All of the description and prescription contained in constraint management is predicated on these assumptions.

Goal and Necessary Conditions

The first assumption above holds that every system has a goal and a set of necessary conditions that must be satisfied to achieve that goal. (Schragenheim and Dettmer, 2000, ch.2) The philosopher Friedrich Nietzsche once observed that by losing your goal, you have lost your way. Or another way of putting it: *if you don't know what the destination is, then any path will do.*

While this assumption is undoubtedly valid in most cases, there are obviously some organizations that have not expended the time or effort to clearly and unequivocally define what their goal is. And even if they have defined a goal, most have not gone the extra step to define the minimum necessary conditions, or critical success factors, for achieving that goal.

For example, most for-profit companies have something financial as their goal. Goldratt contends that the goal of for-profit companies is to "make more money, now and in the future." (Goldratt, 1990, p. 12) Another way of saying this is *profitability*. This, of course, would not be an appropriate goal for a government agency, such as the Department of Defense or Department of Education. Non-financial goals would have to be developed for such agencies. But it works quite well for most companies engaged in commercial business.

However, having profitability as a goal isn't enough. For any organization to be profitable, and for those profits to consistently increase, there is a discrete set of necessary conditions it must satisfy. Some of these will be unique to the industry that the company is in, others will be generic to all for-profit companies. But one thing that all organizations will have in common: there will be very few of these necessary conditions, maybe fewer than five.

Figure 1

Necessary conditions are critical success factors. They are actually required to achieve the goal. For instance, customer satisfaction is unquestionably essential to continued progress toward a financial goal. Employee satisfaction might be considered necessary for achieving the goal. Including necessary conditions like these as part of the goal hierarchy gives it credibility, identifying it as something that is not just temporary but must be satisfied throughout the lifetime of the organization. Figure 1 illustrates a typical goal/necessary condition hierarchy.

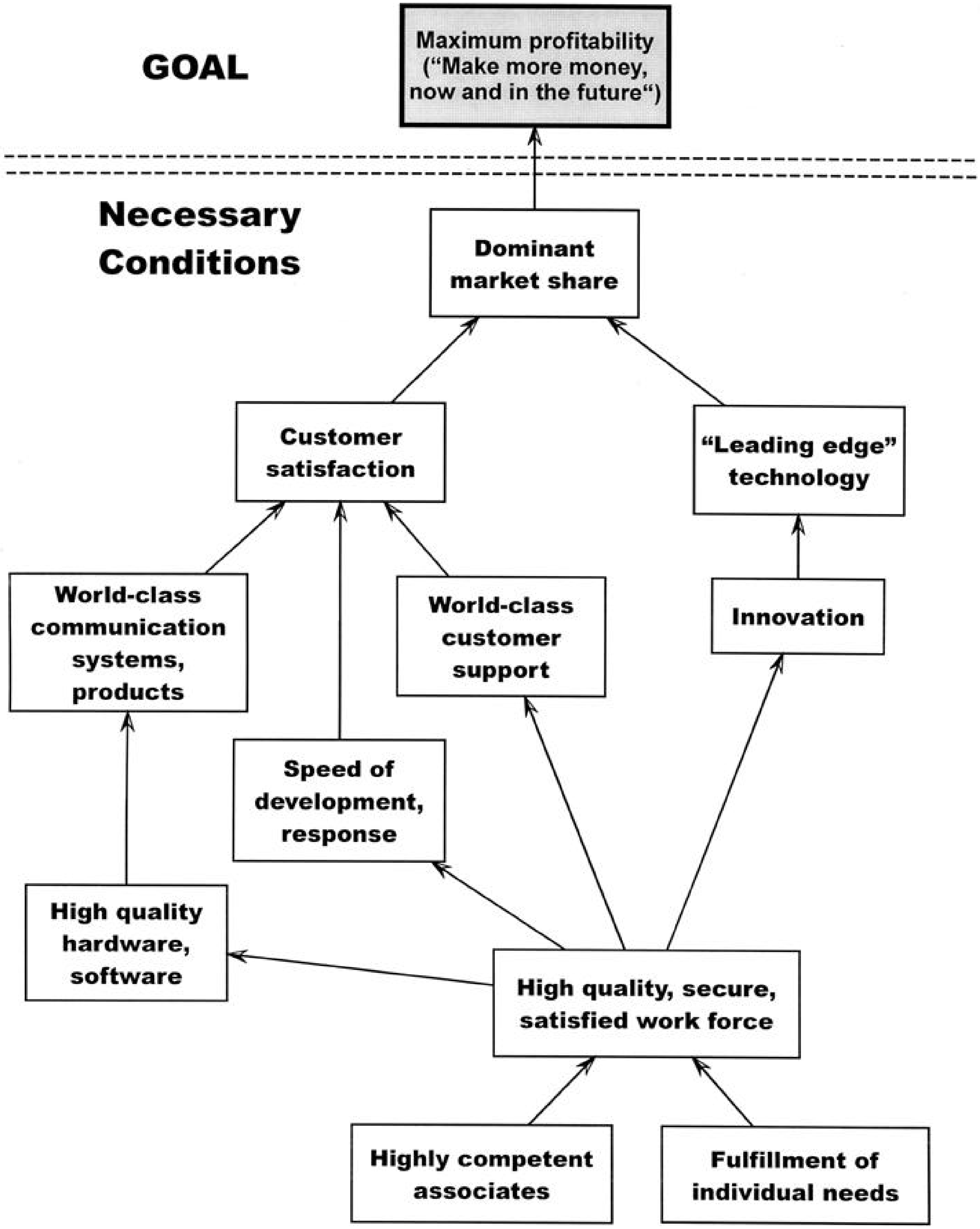


Figure 1 Hierarchy of Goal and Necesary Conditions

Necessary conditions differ from the goal. While the goal itself usually has no limit (it's normally worded in such a way that it's not likely ever to be fully realized), necessary conditions are more finite. They might be characterized as a "zero-or-one" situation: It's either there or it isn't (a "yes-or-no" state). For example, a for-profit organization might want to make as much money as it can—no limits. But employee security and satisfaction, as necessary conditions, should be established at a well-defined minimum level. A for-profit company's goal can't expect to satisfy its employees without limit, but the organization should recognize the need to achieve a certain level of employee security and satisfaction as one minimum requirement for achieving the goal. This is not to say that all necessary conditions are zero-or-one in nature. Some, such as customer satisfaction, can be increased, and doing so can be expected to improve progress toward the goal. But even such variable necessary conditions have practical limits.

The importance of identifying a system's (organization's) goal and necessary conditions is that they become the standard by which all results are judged and all contemplated decisions are evaluated. Did yesterday's, or last month's, actions better satisfy a necessary condition or contribute to realizing the goal? If so, the organization knows that it's making progress in the right direction. Can we expect the decisions contemplated today, or next week, to advance the company toward its goal, or to satisfy a necessary condition? If so, the decision is a good one from the system perspective.

The Five Focusing Steps

Once the necessary conditions are established, constraint theory prescribes applying five "focusing steps" in order to continuously proceed inexorably toward satisfying those necessary conditions. (Goldratt, 1986, p. 307)

Goldratt created the five focusing steps as a way of making sure management "keeps its eye on the ball"—what's really important to success: the system constraint. In one respect, these steps are similar to the Shewhart Cycle: plan-do-check/study-act (Deming, 1986). They constitute a continuous cycle. You don't stop after just one "rotation." The Five Focusing Steps (Schragenheim and Dettmer, 2000, ch. 2) are:

1. Identify. The first step is to *identify the system's constraint.* What limits system performance now? Is it inside the system (a resource or policy) or is it outside (the market, material supply, a vendor... or another policy)? Once the system constraint is identified, if it can be broken without much investment, immediately do so, and revert to the first step again. If it can't be easily broken, proceed to the second step.

2. Exploit. *Decide how to exploit the system's constraint.* "Exploit" means to "get the most" out of the constraining element without additional investment. In other words, change the way you operate so that the maximum financial benefit is achieved from the constraining element. For example, if the system constraint is market demand (not enough sales), it means catering to the market so as to win more sales. On the other hand, if the constraint is an internal resource, it means using that resource in the best way to maximize its marginal contribution to profit. This might mean process quality improvement, re-engineering the flow of work through the process, or changing the product mix. Exploitation of the constraint should be the kernel of tactical planning—ensuring the best performance the system can draw *now*. For this reason, the responsibility for exploitation lies with the line managers who must provide that plan and communicate it, so that everyone else understands the exploitation scheme for the immediate future.

3. Subordinate. Once the decision on how to exploit the constraint has been made, *subordinate everything else to that decision*. This is, at the same time, the most important and the most difficult of the five focusing steps to accomplish. Why is it so difficult? It requires everyone and every part of the system not directly involved with the constraint to subordinate, or "put in second place," their own cherished success measures, efficiencies, and egos. It requires everyone, from top management on down, to accept the idea that excess capacity in the system at most locations is not just acceptable— it's actually a good and necessary thing!

Subordination formally relegates all parts of the system that are not constraints (referred to as "non-constraints") to the role of supporters of the constraint. This can create behavioral problems at almost all levels of the company. It's very difficult for most people to accept that they and/or their part of the organization aren't just as critical to the success of the system as any other. Consequently, most people at non-constraints will resist doing the things necessary to subordinate the rest of the system to the constraint. This is what makes the third step so difficult to accomplish.

What makes the constraint more critical to the organization is its *relative weakness*. What distinguishes a non-constraint is its *relative strength*, which enables it to be more flexible. So the current performance of the organization really hinges on the weak point. While the other parts of the system could do more, because of that weak point *there is no point* in doing more. Instead, the key to better performance is wisely subordinating the stronger points so that the weak point can be exploited in full.

Subordination actually redefines the objectives of every process in the system. Each process is supposed to accomplish a mission that's necessary for the ultimate achievement of the goal. But among processes there may be conflicting priorities, such as competition for the same resources. Subordinating non-constraints actually focuses the efforts of every process on truly supporting the goal of organization. It allows the constraint to be exploited in the best way possible.

Consider a raw material warehouse. What is its objective? The storing and releasing of material is needed as a "bridge" between the time materials arrive from the vendors and the time the same materials are needed on the production floor. When a specific work center is the constraint, any materials needed by that particular work center should be released precisely at the required time. If market demand is the only constraint, any order coming in should trigger material release.

However, even if no new orders enter the system, shop foremen often like to continue working, so as to keep their efficiency high. But if the non-constraints in a production system are properly subordinated, material should *not* be released. The material release process must be subordinated to the needs of the system constraint, not to arbitrary efficiency measurements. Maintaining the order in the warehouse is part of the subordination process. Release of materials not immediately needed for a firm order should be treated as a lower priority than the quick release of materials the constraint will soon need to fulfill a definite customer requirement.

Subordination serves to focus the efforts of the system on the things that help it to maximize its current performance. Actions that contradict the subordination rationale should be suppressed.

It's possible that, after completing the third step, the system constraint might be broken. If so, it should be fairly obvious. Output at the system level will usually take a positive jump, and some other part of the system might start to look like a "bottleneck." If this is the case, go back to the first step and begin the Five Focusing Steps again. Identify which new factor has become the system constraint, determine how best to exploit *that* component and subordinate everything else.

4. Elevate. However, if, after completing Step 3, the original constraint is still the system constraint, at this point the best you can be assured of is that you're wringing as much productivity out of it as possible— it's not possible for the system to perform any better than it is without additional management action.

In taking this action, it's necessary to proceed to the fourth step to obtain better performance from the system. That step is to *evaluate alternative ways to elevate the constraint* (or constraints, in the unlikely event that there is more than one). *Elevate* means to "increase capacity." If the constraint is an internal resource, this means obtaining *more time* for that resource to do productive work. Some typical alternatives for doing this might be to acquire more machines or people, or to add overtime or shifts until all 24 hours of the day are used.

If the constraint is market demand (lack of sales), elevation might mean investing in an advertising campaign, or a new product introduction to boost sales. In any case, elevating invariably means "spend more money to make more money."

Notice the use of the word "evaluate" in this step. This word is emphasized for a good reason. From the preceding examples— buying more equipment or adding shifts, or overtime— it should be clear that there's more than one way to skin a cat. Some alternatives are less expensive than others. Some alternatives are more attractive for reasons that can't be measured directly in financial terms (easier to manage, for example). In any case, a choice on the means to elevate will usually be required, so jumping on the first option that you think of might not necessarily be a good idea.

One of the reasons to favor one elevation alternative over another is the identity of the next potential constraint. Constraints don't "go away," per se. When a constraint is broken, some other factor, either internal or external to the system, becomes the new system constraint—albeit at a higher level of overall system performance, but a constraint nonetheless. It's possible that the next potential constraint might be more difficult to manage than the one we currently have— it might reduce the margin of control we have over our system.

It's also possible that different alternatives might drive the system constraint to different locations—one of which might be preferable to the other. Or it could be that dealing with the potential new constraint might require a much longer lead time than breaking the current constraint. In this case, if we decide to break the current constraint, we would want to get a "head start" on the tasks needed to exercise some control over the new constraint.

Ineffective Elevation: An Example

For example, one company involved in the manufacture of solid state circuit boards found its constraint to be the first step in its process: a surface-mount (gaseous diffusion) machine. (Schragenheim and Dettmer, 2000, ch. 2) Without considering which other resource might become the new constraint, they opted to purchase another surface-mount machine. This certainly relieved the original constraint. But the automated test equipment (ATE)— about eight steps down the production line— became the new constraint, and managing the constraint at this location was no easy task. It was more complex to schedule at that point, and it suffered more problems. Moreover, moving the constraint out of the ATE section was even more challenging. Buying more ATE was more expensive than buying additional surface-mount equipment. Finding qualified ATE operators was also more difficult.

In short, it took more time, effort, and money to manage or break the ATE constraint than it did to break the surface-mount constraint. Had the company been able to anticipate that ATE would become the system constraint, they could have chosen to either a) leave the constraint where it was— at the surface-mount machine, or b) begin long-lead time acquisition of ATE and ATE operators to boost the ATE section's capacity *before* increasing the surface-mount capacity. Doing so would have increased system performance, yet preserved the system constraint at a location that was far easier to manage.

Another important factor to consider is return on investment. Once the company described above broke the surface-mount constraint, there was potential to generate more Throughput, but how much? If the ATE's capacity was only slightly more than that of the original surface-mount machine, the company might have gained only a small increase in Throughput as a payback for the cost of the new surface-mount unit. This could become a definite disappointment.

As long as the next constraint poses a substantially higher limit than the existing one, it's probably safe to say that the company did the right thing. Even if exploiting the ATE is more difficult, the increase in Throughput might be worth the aggravation. The ATE could always be loaded a little less, and the company would still realize more money. What's the lesson here? Assessing the real return on investment from an *elevation* action requires an understanding of constraint theory, where the next constraint will be, and how much Throughput will increase before hitting the next constraint. So the "evaluate" part of the elevation step can be extremely important. It's important to know where the new constraint will occur, because it could affect our decision on how to elevate.

How to Determine Where the Next Constraint Will Be

The easiest way to do this is to apply the first three of the five focusing steps "in our heads," before actually elevating for the first time. In other words, identify the next most-limiting factor, inside or outside the system, that will keep the whole system from achieving better performance after the current constraint is broken. Then determine what actions will be necessary to exploit that new constraint in the future, and how the rest of the system will have to act to subordinate itself to the exploitation of the new constraint.

Once this is done, the ramifications of each alternative to elevate should be obvious, and a better-informed decision is possible about which alternative to choose— and it might not be the obvious choice, or the cheapest one!

5. Go Back to Step 1, But Beware of "Inertia"

Even if the *exploit* and *subordinate* steps don't break the system constraint, the *elevate* step very likely will, unless a conscious decision is made to curtail elevation actions short of that point. In either case, after the *subordinate* or *elevate* steps it's important to go back to the first step (*identify*) to verify where the new system constraint is, or to determine that is has not migrated away from the original location.

Sometimes a constraint moves, not as a result of intentional actions, but as a result of a change in the environment. For instance, a change in preferences of the market might drive a company to change its product mix to such an extent that the constraint moves elsewhere. While such external changes don't happen very frequently, it's worth the effort to go back to the first step from time to time, just to verify that what we believe to be the constraint still is, in fact, the system's limiting factor.

The warning about *inertia* says "Don't become complacent." There are two reasons for this. First, when the constraint moves, the actions or policies we put into place to *exploit* and *subordinate* to the "old" constraint may no longer be the best things to do for the benefit of the whole system. If we don't re-evaluate where the new system constraint is, this deficiency would never be noticed. Second, there is often a tendency to say, "Well, we've solved that problem. There's no need to revisit it again." But today's solution eventually becomes tomorrow's historical curiosity. An organization that's too lazy (or distracted by other demands for its attention) to revisit old solutions can be sure that eventually—probably sooner, rather than later— it won't be getting the best possible performance from its system.

TOOLS OF CONSTRAINT MANAGEMENT

Success or failure in any endeavor often relies on the selection and proper use of the right tools. Constraint management is no exception. While the Five Focusing Steps are effective guidelines for the tactical and strategic management of any kind of system, in specific situations the nature of constraints and the problems associated with them call for different tools and procedures. Exploiting a constraint would be done differently in a service environment than in a production process. Subordination would be different in a heavy manufacturing company that produces standardized products than it would be in a small job shop. Wouldn't it be useful to have an aid that could point us toward the right constraint management actions for each situation?

The Logical Thinking Process

With so many different kinds of constraints, and with policy constraints underlying most of them, how can we identify what specific changes we should be working on? Many of these constraints aren't easy to identify. Often, they're not physical, or they're not easy to measure. They sometimes extend beyond the boundaries of production processes alone, although they still affect manufacturing, and sometimes— especially if they're policies— they pervade the whole organization.

To facilitate the analysis of complex systems, Goldratt created a logical thinking process. The thinking process is composed of six logic diagrams, or "trees." (Dettmer, 1997, 1998). It was specifically designed to analyze the policies of an organization and determine which one(s) might constitute a constraint to better performance.

This thinking process is unique from one perspective: it's one of the few (maybe the only) problem-solving methodologies that goes beyond problem identification and solution generation, and into solution verification and implementation planning. The components of the thinking process include:

1. *The Current Reality Tree (CRT).* Designed to help identify the system constraint, especially when that constraint is a policy of some kind. Figure 2 shows an example of a typical Current Reality Tree.

Figure 2

2. *The "Evaporating Cloud" (EC).* A kind of conflict resolution diagram. Helps create breakthrough solutions to resolve hidden, underlying conflicts that tend to perpetuate the constraint. Figure 3 illustrates a typical Evaporating Cloud.

Figure 3

3. *The Future Reality Tree (FRT).* Tests and validates potential solutions. Provides logical verification that a proposed solution will actually deliver the desired results. Figure 4 depicts a Future Reality Tree.

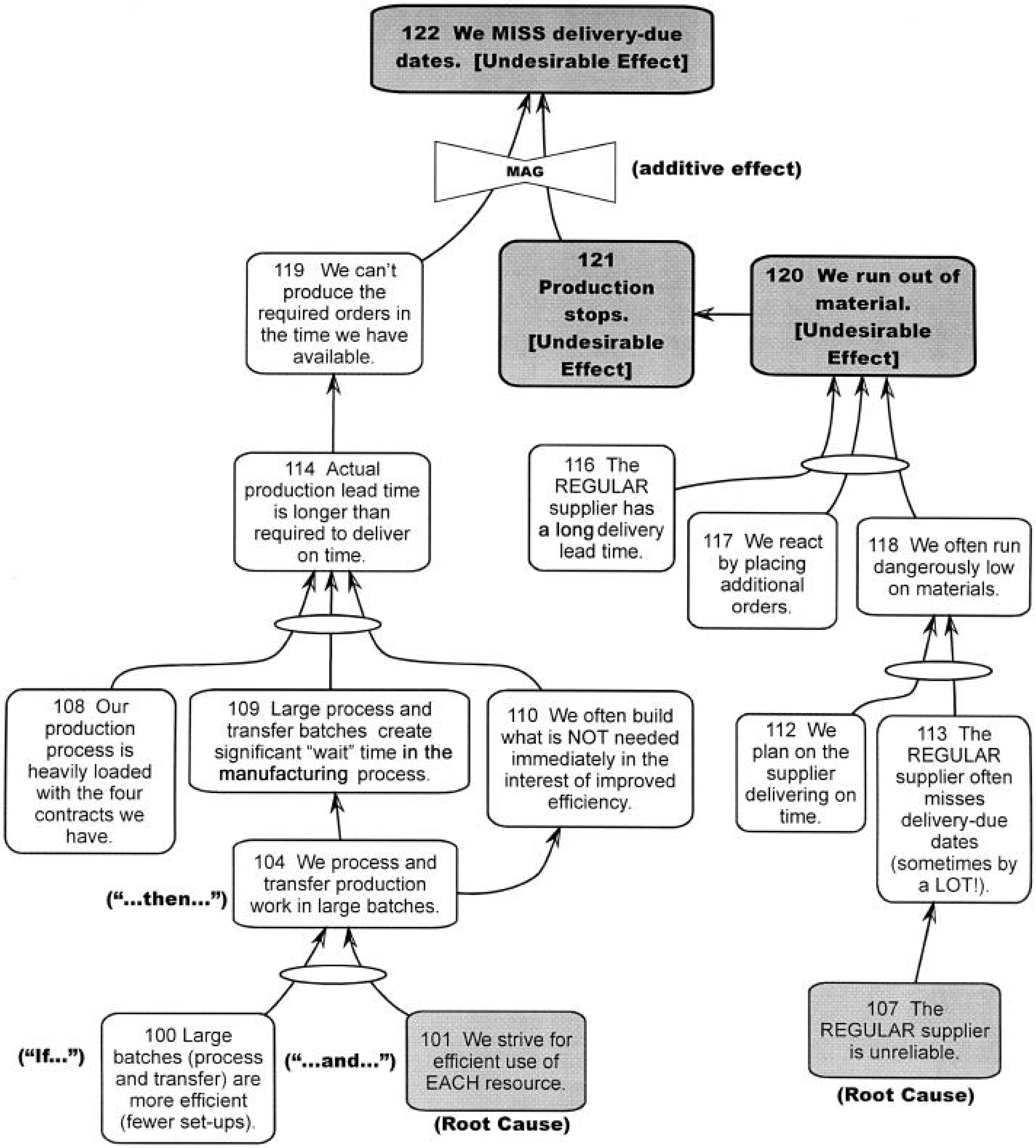
Figure 4

4. *The Negative Branch (NB).* Actually a subset of the FRT. Helps identify and avoid any new, devastating effects that might result from the solution. Figure 5 represents a notional example of a Negative Branch, and how it might have been used to anticipate the disastrous consequences of a very high-profile decision. Notice that this example underscores the fact that application of the thinking process tools is not confined to commercial business situations alone.

Figure 5

5. *The Prerequisite Tree (PRT).* Helps to surface and eliminate obstacles to implementation of a chosen solution. Also time-

Figure 6

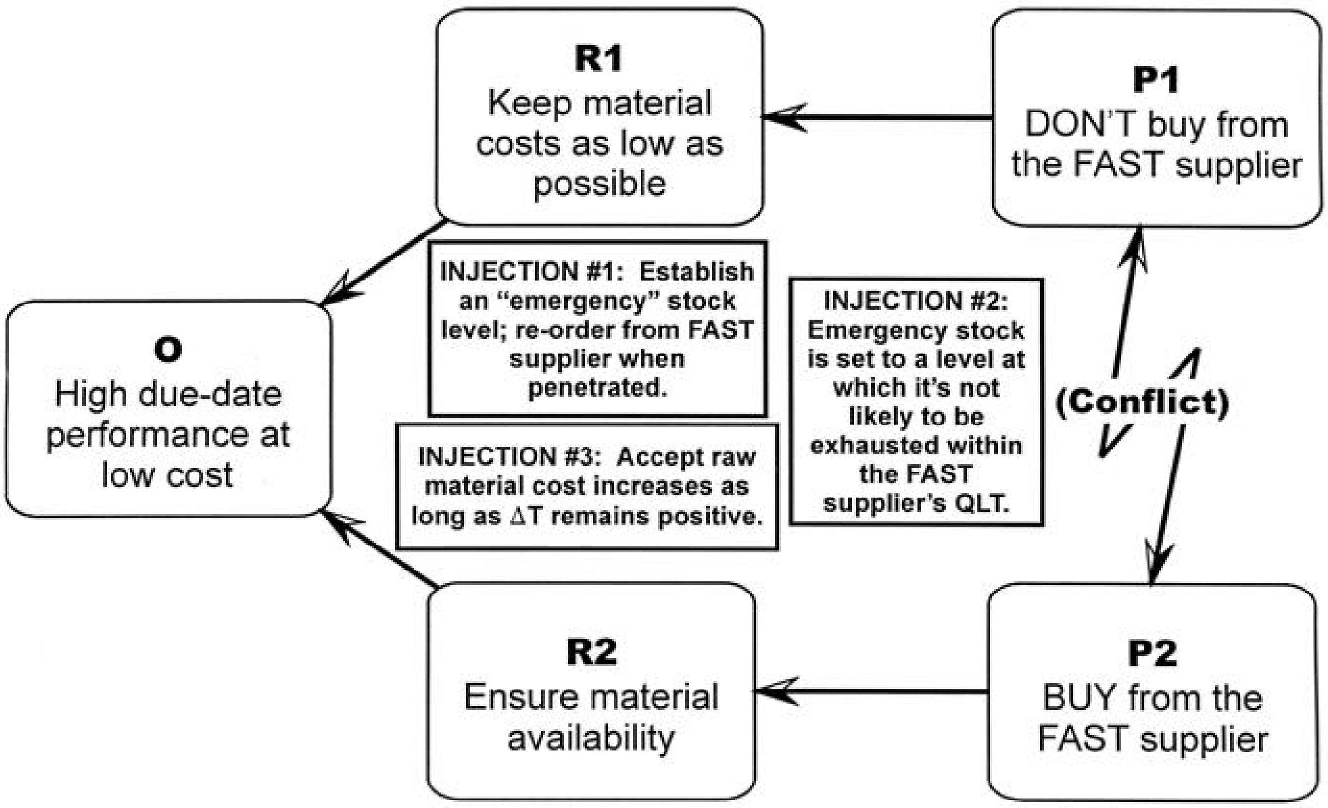


(Adapted from Schragenheim and Dettmer, 2000)

Figure x.2 Current Reality Tree - Manufacturing Example

ASSUMPTIONS:

1. ***The FAST supplier is more expensive than other supplier***
2. ***Buying from the FAST supplier is ALWAYS more costly***
3. ***Purchase costs are significant***
4. ***All cost savings are important***

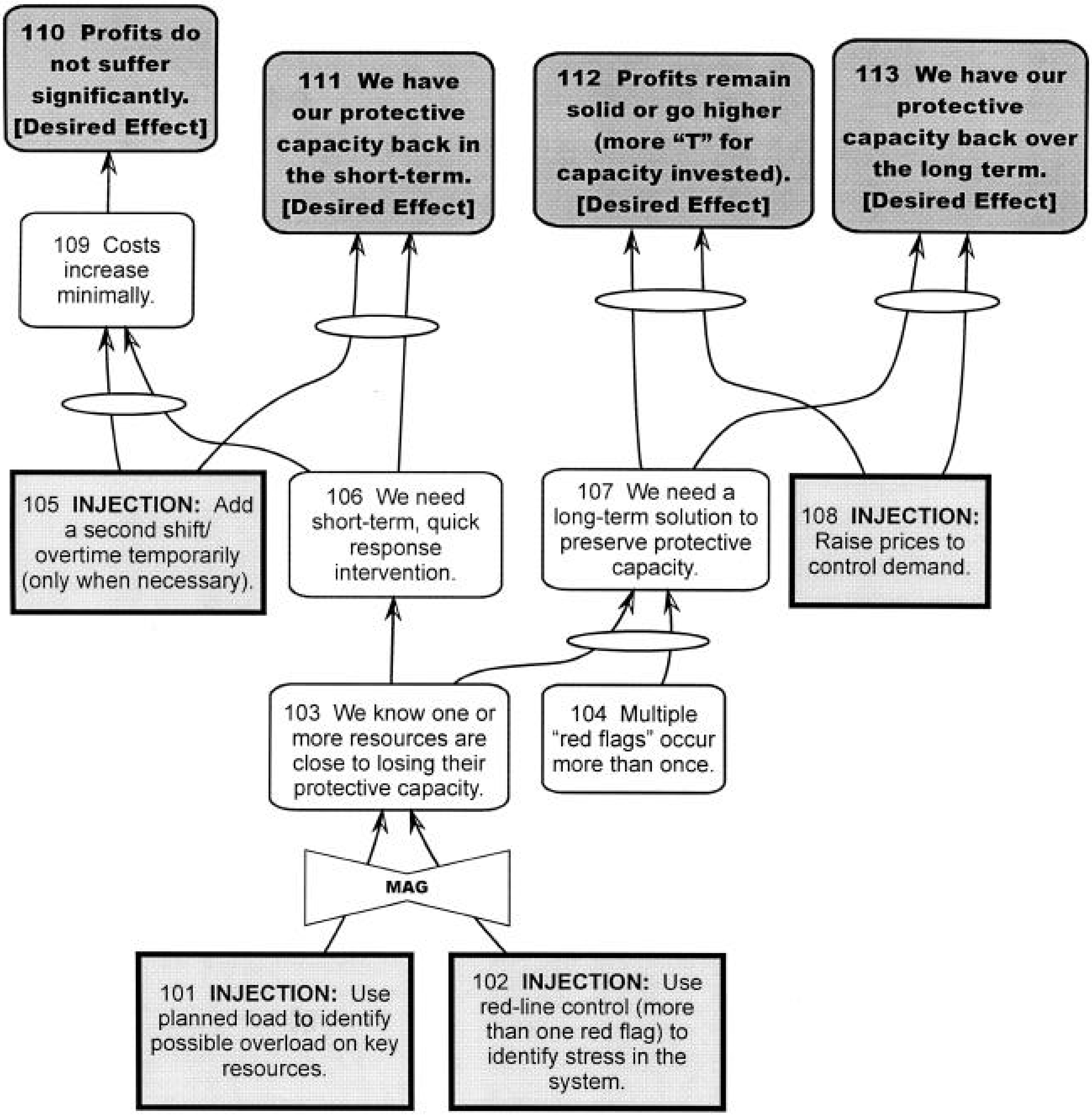


ASSUMPTIONS:

5. ***The regular supplier is unreliable***
6. ***The regular supplier takes too long to deliver***
7. ***We never know about peak demands in time to order from the regular supplier***
8. ***Purchase cost is less important than the cost of a missed delivery***

(Adapted from Schragenheim and Dettmer, 2000)

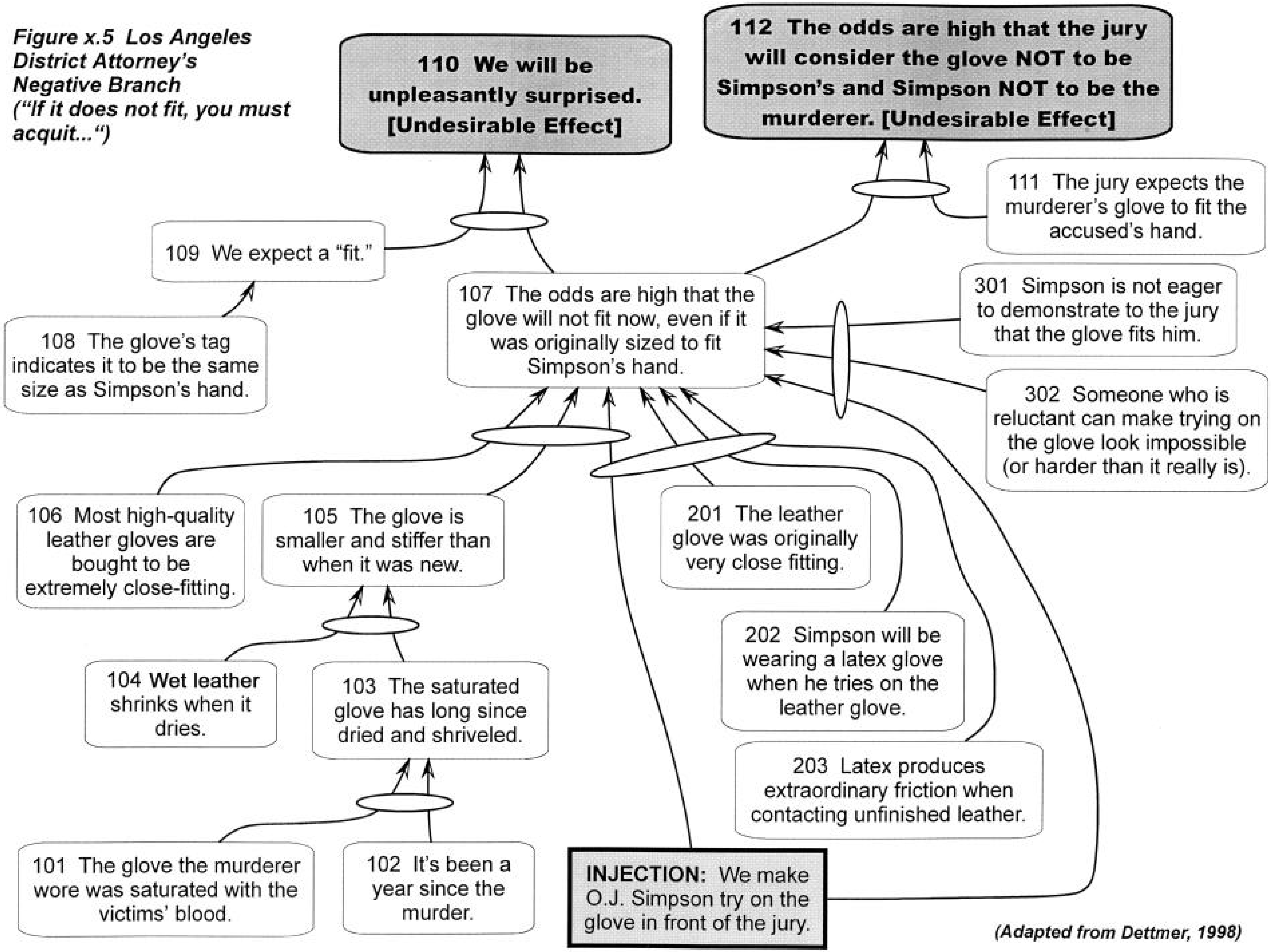
Figure 3 "Evaporating Cloud" - "Unreliable Supplier"



(Adapted from Schragenheim and Dettmer, 2000)

Figure x.4 Future Reality Tree - "Red-Line Control" (manufacturing control)

Figure x.5 Los Angeles District Attorney's Negative Branch ("If it does not fit, you must acquit...")



(Adapted from Dettmer, 1998)

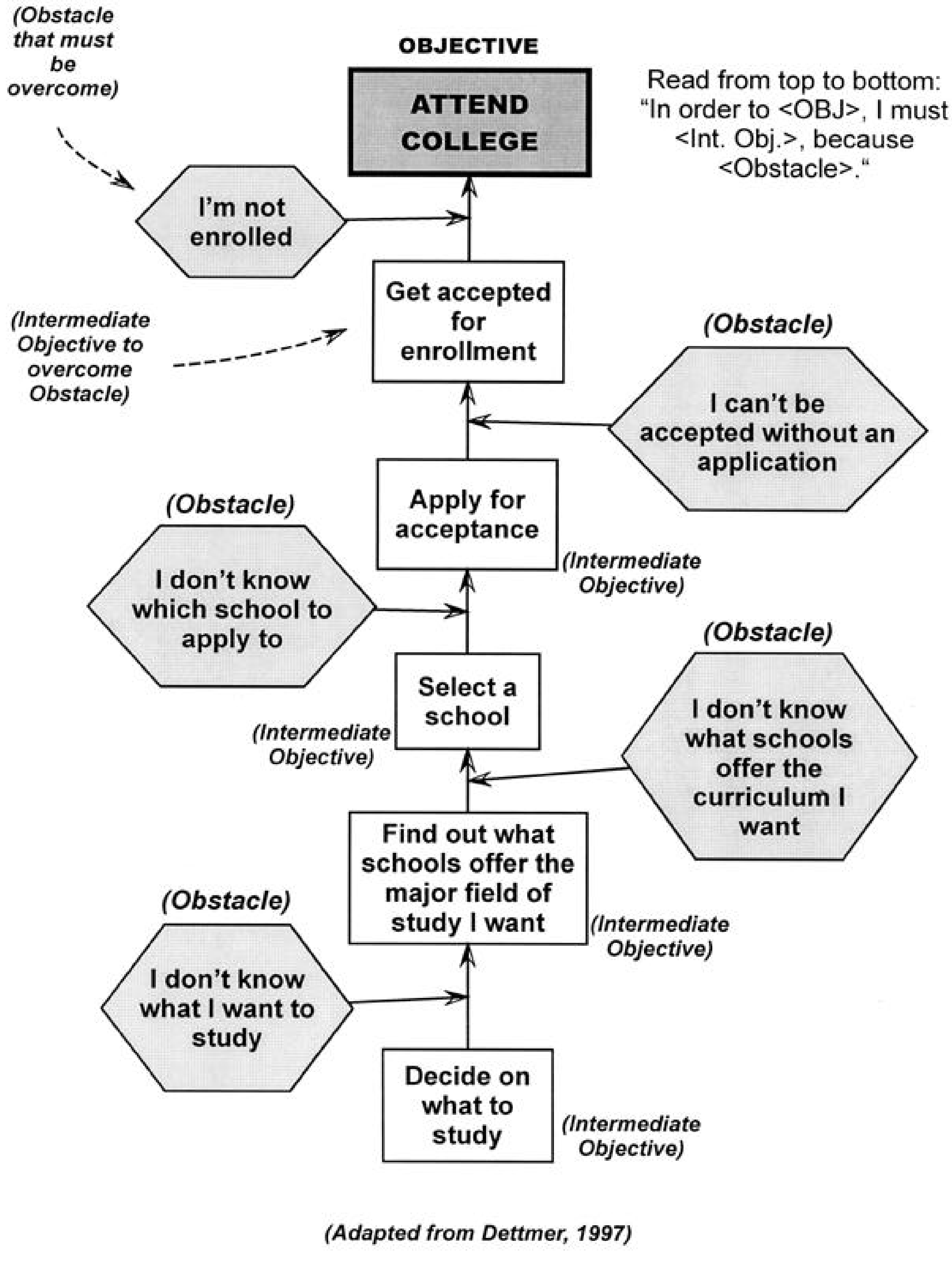


Figure 6 Prerequisite Tree (Example)

sequences the actions required to achieve the objective. Figure 6 shows a typical Prerequisite Tree.

6. The Transition Tree (TT). Can facilitate the development of step-by-step implementation plans. Also helps explain the rationale for the proposed actions to those responsible for implementing them. This can be especially important when those charged with executing a plan are not the same people who developed it. Figure 7 contains a typical Transition Tree. Either the Transition Tree or the Prerequisite Tree can form the basis of a project activity network for implementation of change.

Figure 7

Four of these trees—the Current Reality Tree, Future Reality Tree, Negative Branch and Transition Tree are cause-and-effect trees. They're read using "If... then...". The Evaporating Cloud and Prerequisite Tree are referred to as "necessary condition" trees. They're read a little differently, using "In order to have... we must...".

These tools are specifically designed to help answer the three major questions inherent in the first three of the five focusing steps:

What to change?
What to change *to?*
How to *cause* the change?

"Drum-Buffer-Rope" Production Scheduling

Probably the best-known of the constraint management tools developed by Goldratt is called "Drum-Buffer-Rope" (DBR). The origin of this name dates back to the analogy Goldratt and Cox used in *The Goal* (Goldratt, 1986) to describe a system with dependencies and statistical fluctuations. The analogy was a description of a boy scout hike. The drum was the pace of the slowest boy scout, which dictated the pace for the others. The buffer and rope were additional means to ensure all the boy scouts walked at approximately the pace of the slowest boy.

Goldratt and Fox, in *The Race* (Goldratt and Fox, 1986), describe in detail the manufacturing procedure that stems from the concepts of a drum, buffer and rope originally introduced through the boy scout hike. The DBR method provides the means for synchronizing an entire manufacturing process with "weakest link" in the production chain. Figure 8 illustrates the DBR concept.

Figure 8

The "Drum." In a manufacturing or service company, the "drum" is the schedule for the resource or work center with the most limited capacity: the capacity-constrained resource (CCR). The reason the CCR is so important is that it

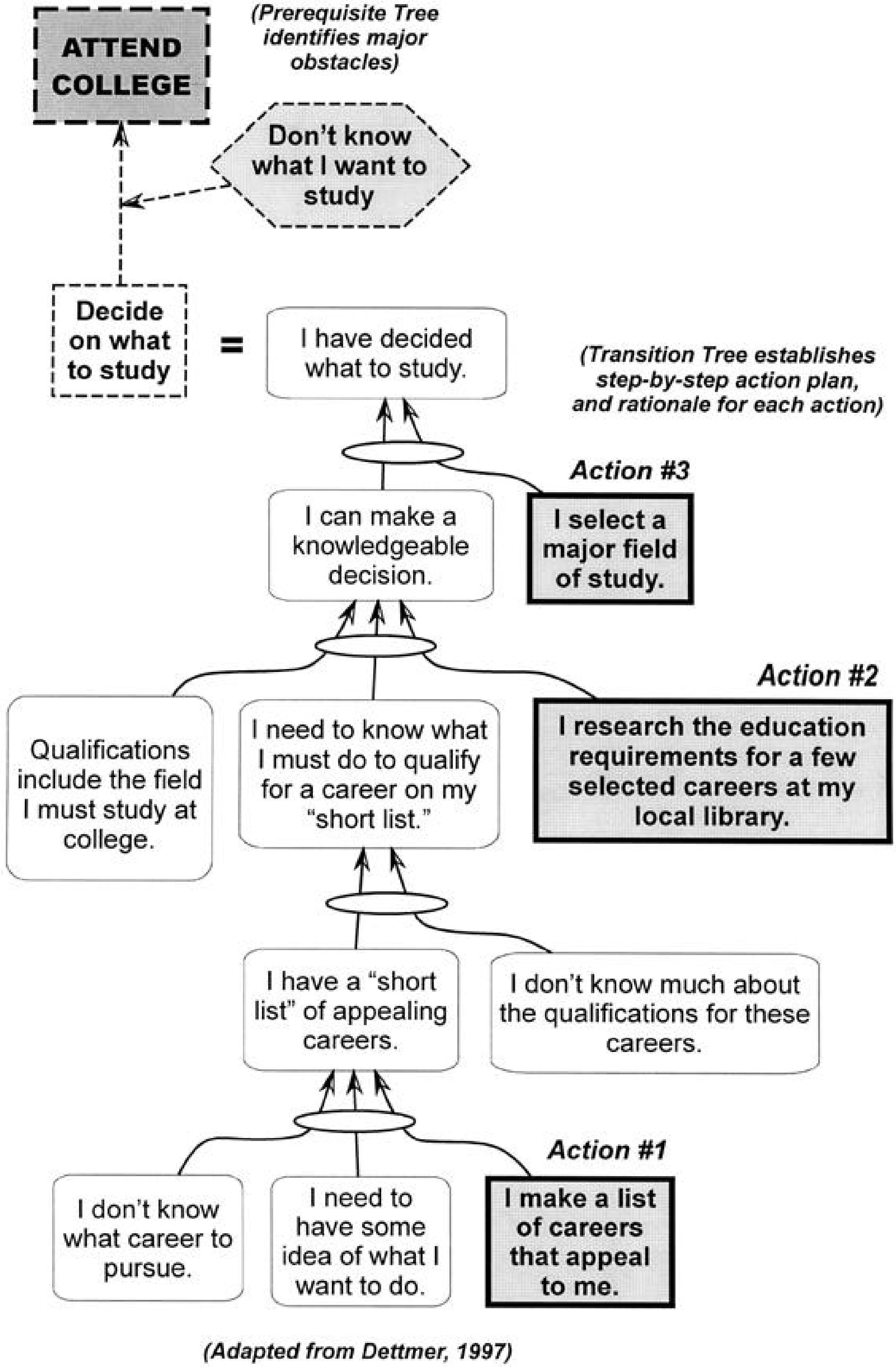
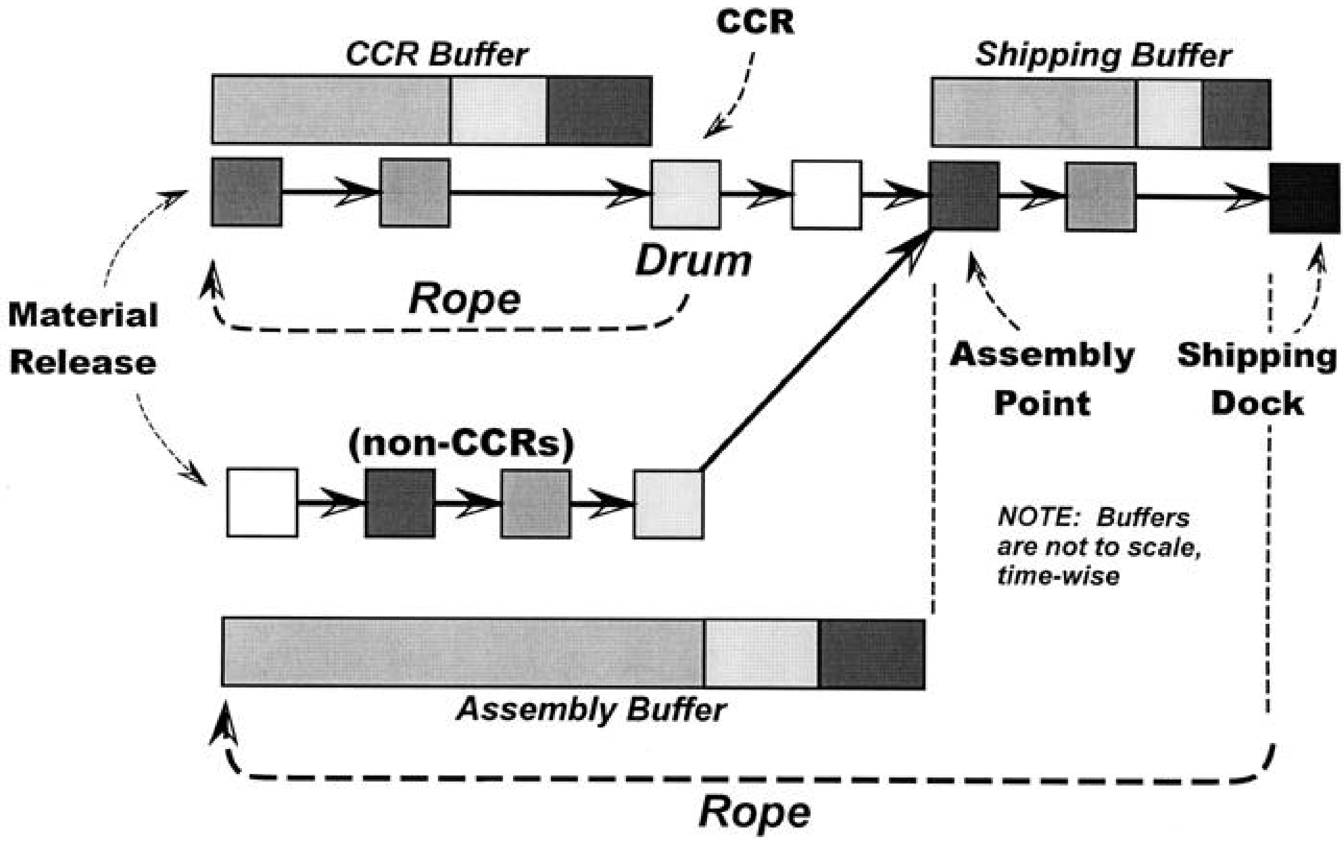


Figure 7 Transition Tree (Example)



(Adapted from Schragenheim and Dettmer, 2000)

Figure 8 Basic DBR Concept

determines the maximum possible output of the entire production system. It also represents the whole system's output, since the system can't produce any more than its least-capable resource.
The "buffer" and the "rope" ensure that this resource is neither starved for work nor overloaded (causing backlogs). In constraint management, buffers are composed of *time*, not *things*.

The "Buffer." Starvation can result from upstream process variability, which might delay the transfer of work-in-process beyond its expected time. To ensure a CCR is not starved for work, a buffer time is established to protect against variability. This is a period of time in advance of the scheduled "start processing" time that a particular job arrives at the CCR. For example, if the CCR schedule calls for this valuable resource to begin processing a particular work order at 3:00 PM on Tuesday, the material for that job might be released early enough to allow all preceding processing steps to be completed at 3:00 PM on Monday (a full work day ahead of the time required). The buffer time serves to "protect" the most valuable resource from having no work to do—a serious failing, since the output of this resource is equivalent to the output of the entire system.

It should be noted that only critical points in a service or production process are protected by buffers. (Refer to Figure 8) These critical points are the capacity-constrained resource, any subsequent process step where assembly with other parts occurs, and the shipping schedule. Because the protection against variability is concentrated only at the most critical places (and eliminated everywhere else in the process), actual lead time can be shortened considerably, sometimes by 50 percent or more, without compromising due-date reliability. Shorter lead times and higher delivery reliability are important service characteristics that customers often look for.

The "Rope." The rope is constraint management's safeguard against overloading the CCR. In essence, it's a material release schedule that prevents work from being introduced into the system at a rate faster than the CCR can process it. The rope concept is designed to prevent the backlog of work at most points in the system (other than the planned buffers at the critical protected points). This is important, because work-in-process queues are one of the chief causes of long delivery lead times.

When the entire Drum-Buffer-Rope concept is applied, delivery reliability of 100 percent is not an unreasonable target, and actual lead time reductions of 70 percent are common. Bethlehem Steel's Sparrows Point plant increased delivery reliability from 49 percent to nearly 100 percent while they reduced actual lead times from 16 weeks to about four weeks. (Dettmer, 1998, ch. 1)

Critical Chain

Another valuable asset in the constraint management toolbox is called "critical chain." Also the title of a book by Goldratt (Goldratt, 1997), the critical chain concept provides an effective way to schedule project activities by effectively accommodating uncertainty and resolving simultaneous needs (contentions) for the same resource. Critical Chain constitutes the application to one-time projects of the same principles that DBR applies to repetitive production. The result of applying Critical Chain scheduling and resource allocation is a higher probability of completing projects on time, and, in some cases, actually shortening total project duration. Originally applied to the management of a single project, the Critical Chain method has been expanded to multi-project environments, based on the concept of the "drum," described in ***Drum-Buffer-Rope,*** above.

Since projects aren't quite the same as repetitive production, some differences in employing Critical Chain project planning are inevitable. But the concepts are much the same as those of DBR. What distinguishes Critical Chain from PERT/CPM and other traditional project management approaches?

First, Critical Chain recognizes and accounts for some human behavioral phenomena that traditional project management methods don't. (Leach, 2000; Newbold, 1998) These phenomena include:

1. The tendency of technical professionals to "pad" their time estimates for individual tasks, in an effort to protect themselves from late completion.

2. The so-called "student syndrome"—waiting until the last minute to begin work on a task with a deadline.

3. Parkinson's Law (ensuring that an activity consumes every bit of the estimated time, no matter how quickly the associated tasks can actually be completed).

4. Multitasking—the tendency of management to assign people more than one deadline activity simultaneously. Multitasking can create a devastating effect. Project personnel switch back and forth between several tasks, causing "drag" in all of them. The result is that other resources that depend on these task completions for their inputs are delayed. Delays "cascade" when several simultaneous projects are involved.

The delays that result from this kind of behavior are caused by a flawed management assumption: *The only way to ensure on time completion of a project is to ensure that EVERY activity will finish on time.* This common management belief prompts management to attribute unwarranted importance to meeting scheduled completion time for each separate activity in a project.

People then pad their estimates of task times to ensure they can get everything done in time. But at the same time, they try to *not* finish much earlier than their own inflated estimation, so as not to be held to shorter task times on subsequent projects. All of these “human machinations” cause a vicious circle, dragging projects out longer and longer, but the reliability of meeting the original schedules isn’t improved.

Second, to solve this problem, Critical Chain takes most of the protective time out of each individual activity and positions some of it at key points in the project activity network: at convergence points and just ahead of project delivery. Since accumulating protection on an entire chain is much more effective than protecting every activity, only half of the aggregated “protective pad” extracted from individual activities is put back in at the key locations. The rest can contribute to earlier project completion. In traditional project execution, if protective time in a specific activity isn’t used, it’s lost forever—unusable by later activities that might need more protection than they were originally assigned. This formerly “lost time” is, in many cases, usable in Critical Chain.

Third, Critical Chain devotes more attention to the availability of critical resources when they’re needed for specific activities. Leveling the resources on any single project is mandatory. The Critical Chain is really the longest sequence in the project that considers *both* dependent, sequential activity links *and* resource links. The critical path reflects only the sequential linking of dependent tasks.

The key elements of Critical Chain Project Management include:

Figure 9

*The Critical Chain.* The set of tasks that determines project duration, considering both task precedence *and* resource dependencies. (Newbold, 1998) The longest sequence of dependent activities is the “critical path” in a PERT/CPM approach. But when the duration of this sequence is adjusted for optimum resource availability (resource leveling), the whole definition of the critical path becomes irrelevant. What results is a different time duration, based on resource usage. This is the critical chain. (Refer to Figure 9) Because the critical chain now constitutes the constraint that determines the earliest date that a project can finish, it’s crucial to monitor progress along the critical chain, because it reflects the progress of the entire project.

*The Project buffer.* A project buffer is established at the end of the final activity on the critical chain, and before the required delivery date. It’s designed to protect against extreme variability and uncertainty that may impact the critical chain.

*Feeding buffers.* Each activity (or sequence of activities) that feeds the critical

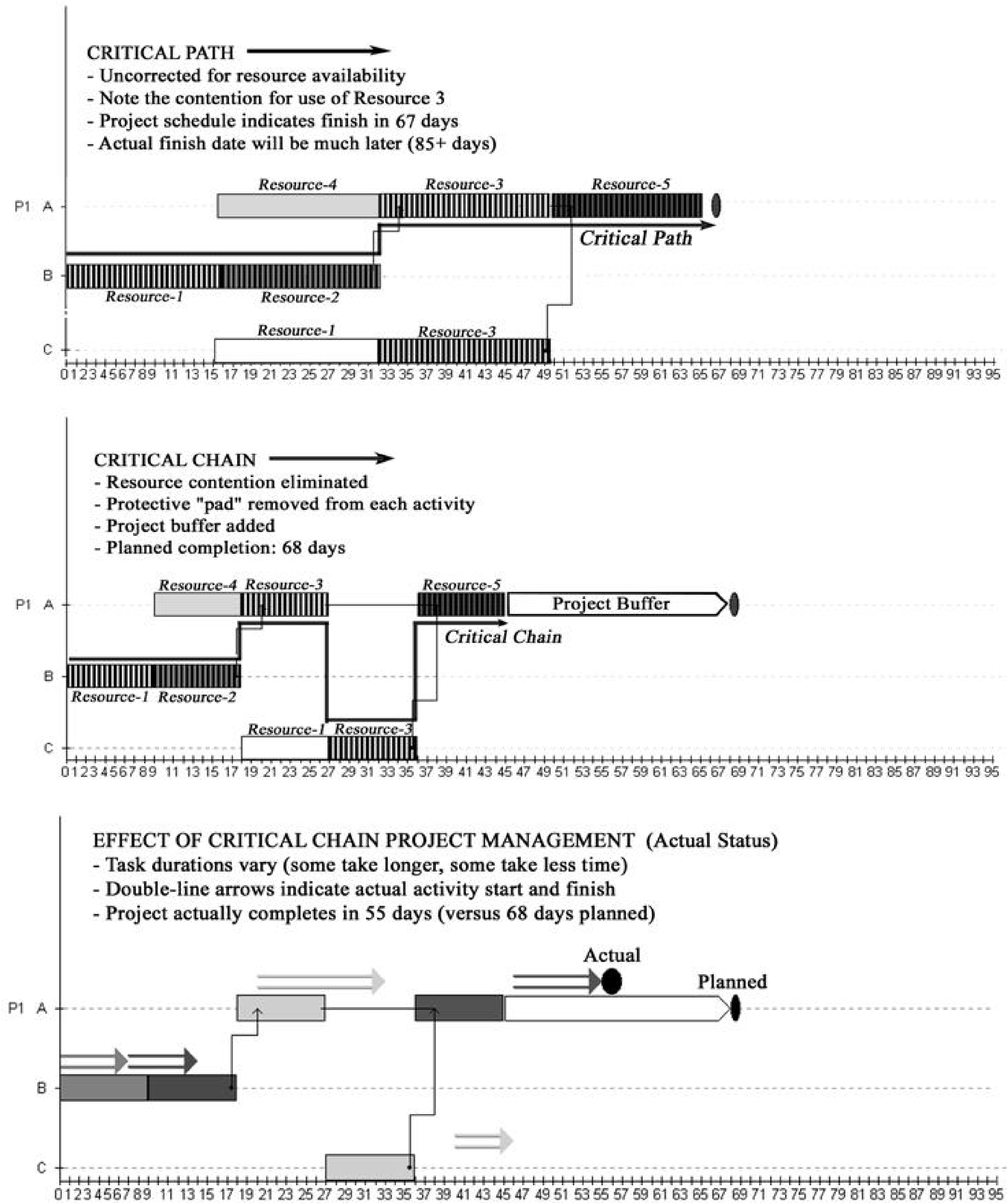


Figure 9 Critical Chain Versus Critical Path

chain is buffered with some reasonable amount of time to protect against variability in that particular task, or sequence of tasks. Feeding buffers protect the critical chain from delays occurring in activities that are not on the critical chain. Variability is *all* that these buffers protect against. They don't protect against *multitasking:* the tendency of organizations to assign technical personnel tasks for different projects or obligations simultaneously. In fact, one of the determinants of success or failure in applying critical chain is the willingness of management to move away from multitasking.

*Buffer Management.* A means of control that, at any time in the project life, affords project managers the opportunity to determine how much of various buffers have been used and to take action as soon as the project appears to be in danger of exceeding its scheduled time. Task completion delays will cause "penetration" into buffers, but warning of this happening comes so much earlier that it's often possible to prevent schedule overruns sooner, with considerably less (or less extreme) corrective action. Monitoring the buffers, especially the project buffer, results in a higher probability that the project will complete on time.

*The Drum.* This concept, similar to that of DBR, applies only to multi-project situations. The drum concept requires choosing one of the most heavily loaded resources as the "drum" and staggering multiple projects according to the availability of that resource.

The ultimate effects of Critical Chain project management is higher delivery due date reliability, more frequent earlier delivery, less "crashing," and conservation of project costs. Buffer management, in particular, provides much better focused information on current project status.

CONSTRAINT MANAGEMENT MEASUREMENTS

One of the unique contributions of the Theory of Constraints to the management body of knowledge is the measurements used to assess progress toward the system's goal. Goldratt recognized some deficiencies inherent in traditional measurement systems and conceived of a different—more reliable—way to measure results and evaluate decisions.

Dilemma: System or Process?

The measurement issue harks back to the earlier discussion about systems versus processes and the fallacy of assuming that the sum of local efficiencies is the system optimum. Traditional rationale maintains that achieving the highest possible productivity in every discrete function of the system equates to good management. Productivity is typically represented as the ratio of outputs to inputs. These inputs and outputs are sometimes expressed in financial terms. Managers often spend inordinate time chasing higher productivity for their own departments, without much concern for whether the whole system benefits or not. This underscores the heart of the problem: How can we be sure that the decisions we make day-to-day truly benefit the system as a whole. In other words, how can local decisions be related to the global performance of the company?

This is not necessarily an easy question to answer. Consider yourself a production manager for a moment. A sales manager comes to you and asks you to interrupt your current production run (that is, break a setup) to process a small but urgent order for a customer. How will doing what the sales manager wants affect the company's bottom line? What will it cost to break the setup (both financially and to the production manager's productivity figures)? How much will it benefit the company? Or the production department? These are not easy questions to answer, yet throughout many companies people are called upon to make such decisions daily.

New Financial Measures

Assuming that a company's goal is to make more money, Goldratt conceived of three simple financial measures to ensure that local decisions line up effectively with this goal. These measures are easy to apply by anyone at virtually any level of a company: *Throughput*, *Inventory* or *Investment*, and *Operating Expense*. (Goldratt, 1990, pp. 19-51).

Throughput (T) is defined as the rate at which a system generates money through sales. (Goldratt, 1990, p.19). Another way to think about it is the marginal contribution of sales to profit. Throughput can be assessed for the entire company over some period of time, or it can be broken out by product line, or even by individual unit of product sold. Mathematically, Throughput equates to sales revenue minus variable cost.

$$T = SR - VC$$

Inventory (I) is defined as all the money the system invests in purchasing things it intends to sell (presumably after adding some value to them). (Goldratt, 1990, p. 23). Because Goldratt's concept of Inventory includes fixed assets, such as equipment, facilities, and real estate, the term "I" has come to represent "investment" as well, rather than just "inventory" alone. Inventory/Investment certainly includes the materials that the company will turn into finished products or services. But it also includes the assets of a company, which are eventually sold off at depreciated or scrap value and replaced with new assets. This is even true of factory buildings themselves. However, for day-to-day decisions, most managers consider "I" to represent the consumable inventory of materials that will be used to produce finished products or services.

Operating Expense (OE) is defined as all the money the system spends turning Inventory into Throughput. (Goldratt, 1990, p. 29). Notice that overhead is not included in the Throughput formula (or the definition). Overhead, and most other kinds of fixed costs, are included in Operating Expense. Constraint management measurements deliberately segregate fixed costs from the Throughput calculation for a valid reason: allocating fixed costs to units of product sold produces a distorted concept of actual product costs in most day-to-day situations.

For example, let's say you're a small manufacturer of precision-machined parts. You're working on an order for 100 units of a particular part for an original equipment manufacturer. In the middle of this run, the customer calls and asks you to add ten more units to the order. This will increase your production time by 53 minutes. How much more have these additional ten units cost you? As long as you're not backlogged with work, the cost of the extra units is the value of the raw materials alone! You didn't pay any more in salary to the machine operator (he or she works by the hour, not by the piece). In most cases you wouldn't pay any more in electricity costs. You have to turn the lights on for business for the whole day anyway. And the cost of the general manager's company car didn't change just because you produced 10 additional units of the product. The real cost of the increase in production volume is limited primarily to the cost of materials alone.

Labor costs are also considered an Operating Expense, because in almost all cases they are paid by some fixed unit of time, not by the individual unit of product produced. We pay people by the hour, week, month, or year, whether they are actively producing a product for sale or not. Moreover, the capacity to produce a product or service (the resources: people, facilities, equipment, etc.) is obtained in "chunks." It's really difficult to hire six-tenths of a person, or to buy three-quarters of a machine.

So the expenditure of cost for capacity usually comes in sizeable increments—a step function. Products, on the other hand, are normally priced and sold by the unit—smaller steps, perhaps, but closer to a continuous function. All this makes it difficult to attach an accurate allocation of fixed costs to a unit of product. Which means that those who do so obtain a distorted impression of product costs—not a good basis from which to form operating decisions.

For daily management decisions, which we'd like to be able to relate to the system's goal, T, I and OE are much more useful than the traditional organizational success measures of net profit (NP), return on investment (ROI) and cash flow (CF). Yet there has to be a connection between the two types of financial measures. And here it is:

$$NP = T - OE$$

$$ROI = \frac{T - OE}{I}$$

$$CF = T - OE \pm)I$$

Net profit is the difference between Throughput and Operating Expense. Return on investment is net profit (Throughput minus Operating Expense) divided by Inventory/Investment. And cash flow is net profit (Throughput minus Operating Expense) plus-or-minus the change in Inventory.

If Throughput is increased, net profit increases, even if Operating Expense remains the same. If Operating Expense is reduced, net profit also increases (as long as Throughput at least remains constant). If Inventory is reduced, ROI increases, even if there are no changes to Throughput or Operating Expense. These measures relate to operational management decisions much better than net profit and return on investment, keeping daily decisions in more line with the system's goal than abstract efficiency measures such as machine utilization, units produced per day/week, etc.

For example, here's how a manager might use T, I and OE to evaluate a decision he or she is contemplating (Dettmer, 1998, p. 33):

- Will the decision result in a better use of the worst-constrained resource? (i.e., more units of product available to sell in the same or less time)
- Will it make full use of the worst-constrained resource?
- Will total sales revenue increase because of the decision?
- Will it speed up delivery to customers?
- Will it provide a characteristic of product or service that our competitors don't have? (e.g. speed of delivery)
- Will it win repeat or new business for us?
- Will it reduce scrap or rework?
- Will it reduce warranty or replacement costs?
- Will we be able to divert some people to do other work (work we couldn't do before) that we can charge customers for?

If it does any of these things, the decision will improve Throughput.

- Will we need less raw material or purchased parts?
- Will we be able to keep less material on hand?
- Will it reduce work-in-process?
- Will we need less capital facilities or equipment to do the same work?

If the answer is "yes," the decision will reduce Inventory or Investment.

Will overhead go down?
Will payments to vendors decrease?

If so, the decision will decrease Operating Expense.

Let's return to the example mentioned earlier about the production manager faced with a decision whether to break a setup in response to a sales manager's request. Using conventional reasoning, an efficiency-oriented production manager would see his productivity figures suffering because of the time lost to doing the new setup. Inserting this urgent order would also disrupt a formal schedule, slipping every subsequent order's scheduled starting and finishing times.

But a constraint-oriented manager would look at it a little differently. His or her first question would be, "Am I internally constrained by a shortage of resources?" The answer should be fairly obvious, because the manager would already be aware of the size of the backlog, if any.
If the answer is "no," the production process has excess capacity to be able to accommodate the urgent order without delaying scheduled work. The cost of the additional order is only the raw materials used and possibly the loss of materials from a unit of the job currently on the machine when the setup is broken. The machine operators' time doesn't cost any more. They're paid by the hour, whether they're doing setups or producing products. The direct increase in net profit to the company by agreeing to the sales manager's request would be the Throughput (sales revenue minus variable costs) for the new order. And it's possible that providing this expedited service to a customer might win more repeat business in the future. So as long as the manufacturing process is not internally constrained, the production manager cognizant of constraint theory would probably accept this new job, while a traditional manager might tell the sales person, "Get in line! We operate on a first-in, first-out basis." And this manager's "numbers" would look good at the end of the month—but what would that decision have done for the company?

The use of T, I and OE for making management decisions is not a replacement for generally accepted accounting procedures (GAAP). Those are required, and will probably always be, for external reporting purposes—annual reports to stockholders, securities and exchange filings, tax reporting, etc. But Throughput, Inventory and Operating Expense are considerably easier for most line managers to understand and use in gauging the financial effects of their daily decisions.

The Strategic Implications of T, I and OE

What makes Throughput, Inventory/Investment and Operating Expense even more beneficial is the strategic implications of their application by senior managers and executives. Of course, every company executive wants to improve net profit, return on investment and cash flow. However, in standard accounting T, I and OE are "embedded" in these terms and often difficult to single out, even though they are better managed separately.

Figure 10

Consider the bar graph in Figure 10. It shows three bars representing T, I and OE. Most companies devote extraordinary effort to reducing costs (both fixed and variable). The cost-of-quality concept emphasizes this as a justification for pursuing quality in the first place. Lean manufacturing does, too. The same is true of reducing inventory. In fact, in many corporate strategies, cost saving features high on the priority list.

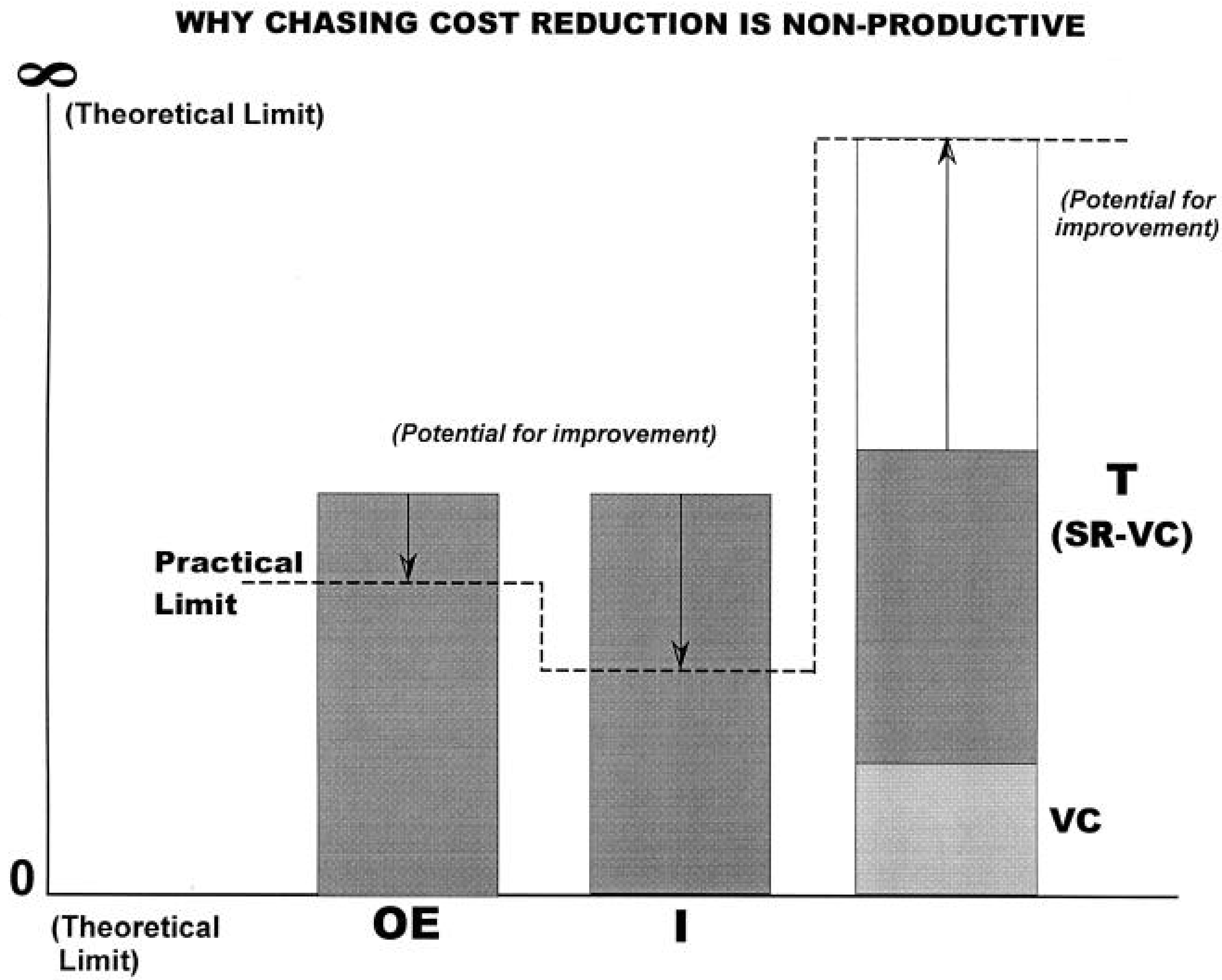
The graph tells us a different story, however. While there might well be savings to be had in these areas (OE and I reductions), these savings have a point of diminishing returns. There's a practical level below which neither Inventory nor Operating Expense can be reduced without hurting a company's ability to generate Throughput. Beyond that point, you're not cutting "fat" anymore, you're cutting "muscle." And the truth is that most managers can't really say where that point lies. Moreover, many companies have been engaged in cost reduction efforts for so long that all of the OE (and even the variable costs of Throughput) have largely been wrung out of the system. To improve financial performance, there's nowhere else to go... except to Throughput. And look at the Throughput bar in Figure x.9. The improvement potential may not be infinite, but from a practical perspective there's a lot more potential to improve profitability by increasing T than there ever will be by reducing I and OE. So what does it make sense to prioritize: Rearranging the deck chairs on the Titanic, or steering away from the iceberg? In summary, not many companies can "save their way" to prosperity.

Using T, I and OE is variously referred to as Throughput accounting, constraint accounting, or Throughput-based decision support. They aren't rigorous enough to replace GAAP for standard accounting needs. But they're simpler and usually more effective for decision-making than traditional management accounting. More details on how to use T, I and OE may be found in Goldratt (1990); Noreen, Smith and Mackey (1995), Corbett (1998), and Smith (1999).

SUMMARY AND CONCLUSION

To summarize, constraint management:

Is a systems management methodology
Separates the "critical few" from the "trivial many"
Emphasizes attention on the critical few factors that determine system



Source: Schragenheim and Dettmer, 2000

Figure 10 Limits to T, I and OE

success

It's based on four underlying assumptions:

1. Every system has a goal and a finite set of necessary conditions that must be satisfied to achieve that goal.

2. The sum of a system's local optima does not equal the global system optimum.

3. Very few variables—maybe only one—limit the performance of a system at any one time.

4. All systems are subject to logical cause-and-effect.

The Theory of Constraints, as conceived by Goldratt, is embodied in the Five Focusing Steps (Identify, Exploit, Subordinate, Elevate, Repeat/Inertia), a set of financial progress measurements (Throughput, Inventory and Operating Expense), and three generic tools (the logical Thinking Process, Drum-Buffer-Rope, and Critical Chain) that can be applied in a variety of organizational situations.

There is considerably more to constraint management and the Theory of Constraints in particular than we've covered here. The references at the end of this chapter provide a good start for deeper inquiry. One of the best sources to begin such an inquiry is Mabin and Balderstone (2000), which is a comprehensive review of all known published literature on constraint theory, and includes brief abstracts of each citation and a statistical analysis of published results.

REFERENCES

Athey, Thomas H. *The Systematic Systems Approach*. NJ: Prentice-Hall, Inc., 1982, p. 12.

Blanchard, Kenneth, and Spencer Johnson. *The One-Minute Manager*. NY: William Morrow & Co., 1982.

Corbett, Thomas. *Throughput Accounting*. Great Barrington, MA: North River Press, 1998.

Deming, W. Edwards. *Out of the Crisis*. Cambridge, MA: MIT Center for Advanced Engineering Study, 1986.

Deming, W. Edwards. *The New Economics for Industry, Government, Education*.

Cambridge, MA: MIT Center for Advanced Engineering Study, 1993.

Dettmer, H. William. Goldratt's *Theory of Constraints: A Systems Approach to Continuous Improvement.* Milwaukee, WI: ASQ Quality Press, 1997.

Dettmer, H. William. *Breaking the Constraints to World-Class Performance.* Milwaukee, WI: ASQ Quality Press, 1998.

Goldratt, Eliyahu M. *The Goal*. Croton-on-Hudson, New York: The North River Press, 1986.

Goldratt, Eliyahu M. *The Haystack Syndrome: Sifting Information Out of the Data Ocean*. Croton-on-Hudson, New York: The North River Press, 1990, p. 53.

Goldratt, Eliyahu M. *Critical Chain*. Great Barrington, MA: The North River Press, 1997.

Goldratt, Elyahu M. and Robert E. Fox. *The Race.* Croton-on-Hudson, New York: The North River Press, 1986.

Leach, Lawrence P. *Critical Chain Project Management.* Boston, MA: Artech House, Inc., 2000.

Mabin, Victoria, and Steven J. Balderstone. *The World of the Theory of Constraints: A Review of the International Literature.* Boca Raton, FL: St. Lucie Press, 2000.

Newbold, Robert C. *Project Management in the Fast Lane: Applying the Theory of Constraints*. Boca Raton, FL: St. Lucie Press, 1998.

Noreen, Eric, Debra Smith, and James T. Mackey. *The Theory of Constraints and Its Implications for Management Accounting.* Great Barrington, MA: North River Press, 1995.

Schragenheim, Eli, and H. William Dettmer. Manufacturing at Warp Speed. Boca Raton, FL: St. Lucie Press, 2000, ch. 2.

Senge, Peter M. *The Fifth Discipline: The Art and Practice of the Learning Organization.* New York: Doubleday, 1990.

Smith, Debra. *The Measurement Nightmare: How the Theory of Constraints Can Resolve Conflicting Strategies, Policies, and Measures.* Boca Raton, FL: The St. Lucie Press, 2000.

Taylor, Frederick W. *Scientific Management.* New York: Harper & Brothers, 1947.

Webster's New Universal Unabridged Dictionary. New York: Barnes and Noble Books, 1989.

