Using Decision Engineering to Achieve Short Predictable Lead Time at Sun Microsystems, Inc.

By

Tom Chavez
Gregory R. Dion
Sun Microsystems, Inc.

Cycle time reduction often requires coordination across a broad range of functions and groups within an organization. This article describes a decision engineering approach that has proved useful in the analysis and tactical implementation of cycle time reduction at Sun Microsystems. First, a simple model called the Decision Chain is introduced to illuminate the interdependencies among decisions taken at different points of a company's value chain. Second, a Complexity Breakdown Model that helps decision-makers view the elements of a cycle time reduction problem in broad terms and identify the places where they can exert the most leverage to achieve cycle time breakthroughs is described. These tools, taken together, can help guide cycle time reduction efforts in other companies.

As others have observed in this forum, cycle time reduction (CTR) requires the coordination and sharing of information (Nichols, Frolick, and Wetherbe, 1995). It often centers on the creation and exchange of shared organizational knowledge (Janz, 1996). It can also require the balancing of several competing objectives (Nichols, Retzlaff-Roberts, and Frolick, 1996). Decision analysis (Howard and Matheson, 1983) pays special attention to the creation and exchange of information and knowledge as well as the balancing of disparate objectives. The primary focus of decision analysis, however, is decision-making.

We believe that CTR is a prime area for the application of tools and principles from decision analysis. Corporate executives make decisions about how to allocate precious resources to achieve CTR. Knowledge workers in large companies spend their days participating in processes that seek to convert data into information and information into knowledge, all in the service of decision-making that strikes the right balance among cost, revenue, quality -- and, increasingly, cycle time.

Decision engineering (DE) is the large-scale application of the tools and techniques of decision analysis to formulate and solve complex problems in systems and organizational design (Chavez, 1996). Decision engineering may require sophisticated mathematical analysis. Other times it may simply require the creation and application of simple distinctions to identify the primary elements of a decision problem. Practitioners have frequently concentrated on the mathematics behind decision analysis (DA) and less on the simple concepts that drive its application. In this paper we report on the use of simple, non-mathematical concepts from DA to structure, solve and attack a large-scale problem in cycle time reduction at Sun Microsystems.

The Short Predictable lead time (SPL) team at Sun was formed during the summer of 1996. It was originally given the task of solving a CTR problem that several groups inside the company had previously addressed: how to make order fulfillment lead time, the time elapsed between when a customer requests a product and when he receives it, shorter and more predictable. Previous efforts had not achieved traction inside the company. Management, meanwhile, was
Do the local optima achieved within functional groups support the firm's stated global objective?

Figure 1: Local vs. Global Optimization

becoming frustrated by delays in meeting competitive requirements for fast, predictable product delivery.

SPL's approach was explicitly cross-functional. Previous efforts had concentrated on modifying individual sub-processes in specific functional groups without taking an aerial view of organizational interdependencies that shape Sun's total cycle time performance. Such approaches failed to analyze the connection between upstream marketing decisions, for example, and their resulting effects on the decisions that operations groups make as they attempt to improve lead time. We developed a tool we call the Decision Chain to help executives visualize their decisions and the connections among them, along with their ultimate effects on customers.

The Decision Chain also helped executives internalize the need for greater cross-functional coordination to address Sun's cycle time challenges.

An important theme in CTR at Sun has been the analysis and management of complexity. As computer companies deliver more products to increasingly demanding customers under severe demand volatility and rising time-to-market, time-to-volume constraints, the complexities of doing business in a high-tech industry often appear overwhelming. The topic is especially relevant here because it can clog early efforts at CTR: Where should we focus attention? What parts of this looming complexity can we reduce, and which parts must we simply live with? What types of complexity act as primary cycle time bottlenecks?

Figure 2: A Sample Decision Diagram for the Decision About Whether to Build a Widget
Using Decision Engineering to Achieve Short Predictable Lead Time

SPL's Complexity Breakdown Model provides a simple decision-analytic means of identifying key elements of complexity in the business environment. The SPL team used the model to identify, prioritize, and exploit opportunities for CTR in a sensible, directed manner. For executives with busy schedules and operators who have not had the time to acquire SPL's exhaustive perspective of Sun's overall cycle time challenge, the model also provided a useful means of quickly communicating the thrust of SPL's solution strategy.

Connecting Functional Organizations Using a Decision Chain

The SPL team recognized early on that, to be successful, it would have to depart from previous function-specific approaches to process improvement. While it is always easier to assemble a team to improve or redesign a process within a bounded functional group, frequently those function-specific teams end up creating locally optimal solutions that do not support the company's global objective. Such behavior leads to situations where everybody is convinced they are doing the right thing -- while "aerial analysis" often indicates otherwise (Figure 1).

Decision analysts use graphical models such as decision diagrams (Howard and Matheson, 1983) and knowledge maps (Howard, 1989) to structure and solve complex decision problems. In the graphical semantics of decision diagrams, squares or rectangles denote decisions, diamonds denote objectives or values, and circles denote probabilistic or other functional quantities. Figure 2 provides an example.

The Decision Chain uses similar graphical semantics to illuminate the decisions of cross-functional participants in an organization. As with the Decision Diagram, we use squares or rectangles to denote decisions. Squares correspond to decisions in specific groups, functions, or value-generating activities. A simple place to begin is the value chain of the manufacturing enterprise, as displayed in Figure 3.

Arcs in a conventional decision diagram denote what is typically called the "no-forgetting"

1The discerning reader will note that the semantics of decision diagrams depart from the conventional semantics of systems diagrams, where diamonds typically denote decision points. Since we are interested in building on previous work in decision engineering, we will adhere to decision analytic conventions in this paper.
condition: an arc from a predecessor decision node into another decision node means that the decision taken at the predecessor is simply known at the successor. Arcs in a decision chain, meanwhile, reflect a much stronger condition: the decision alternatives available at the successor are functionally determined by the alternatives chosen at a predecessor. To describe this stronger type of dependence -- what we term decision dependence -- we use directed arcs whose tips cross the border of the successor decision node as in Figure 3 (Chavez and Spinnler, 1996).

Consider the example in Figure 4. Figure 4 shows how design decisions can limit supplier management's ability to achieve cycle time reductions in component procurement by over-constraining operations' range of decision opportunities.

As an example, consider a Doelling screw that is used to bolt the chassis on Sun's server products. What differentiates it from a Phillips head screw available from any neighborhood hardware store is a spring-like casing that covers its entire length. The screw is an engineering marvel, and in fact, contributes to the reliability and serviceability of the server. The downside is that this screw is custom-made, utilizes patents owned by a single manufacturing firm, and has highly volatile lead times that range between 10 and 45 weeks.

This screw and other components like it demonstrate a design mindset in the engineering groups that can have dangerous effects in downstream operations groups, particularly when it comes to CTR. For example, consider what happens when we increase the use of components like the Doelling screw with the spring-like casing. Factor in the effects of inevitable problems in the supply base for those components. Add to this the complexities of managing such a broad array of risky components and the human errors that naturally result. Are the aggregate risks to supply worth the marginally increased "coolness" of the total product? Sometimes, perhaps. More often than not, however, the risks do not outweigh the benefits.

Decision interdependence often cuts across multiple groups. The negative effects of upstream decisions are not apparent in downstream, functionally isolated groups, and they can frequently spiral out of control. Consider the example in Figure 5.

Figure 5 shows how the locally rational attempt inside marketing to increase total market share can actually increase customer dissatisfaction. While it is difficult to measure the strategic costs of angry, confused customers, the point is that the marketing group, in this case at least, ends up far
away from one of its key, stated objectives. The reason is that they ignored the powerful effects of decision interdependence: They did not analyze -- or internalize -- how their locally rational decision constrains the decision opportunities available to operations, which sits between them and the end customer.

The recognition of decision interdependence and the complexities that come with it was the first big step towards identifying an d implementing the architectural changes needed for large-scale CTR at Sun. For example, the decision dependence shown in Figure 4 has led to the creation of a new methodology for product design called DFL, for Design for Lead time. The basic thrust of DFL at Sun is to get supply specialists more involved in up-front product design so that the resulting hardware products avoid the use of components such as the Doelling screw.

The analysis conveyed in the decision chain in Figure 5 has led to an effort called PCM, for Product Complexity Management. The goal of PCM is to exert more discipline in the management of Sun's products, so that factors of product attractiveness in terms of revenue, cost and “suppliability” contribute to a more balanced assessment regarding which products to keep and which ones to eliminate.

The **Complexity Breakdown Model**

As the SPL team went about its analysis of cycle time bottlenecks at Sun, it gradually came to an unsettling realization: If it were to be successful, it would need to engage executives in an unusually broad, highly coordinated attack.

Process improvements in individual groups would certainly help, but the interdependencies mentioned in the previous section made it necessary for managers in sales, marketing, engineering, supplier management, manufacturing, and logistics to work together in an extremely cooperative fashion.

To get them to collaborate, we first had to get the cross-functional group to understand the problem in terms of their own functional perspectives. We have learned that cycle time solutions are only as good as our understanding of the problem and level of conviction driving the implementation of corrective actions.

In qualitative terms, decision analysis is about the creation and management of distinctions to structure problems and their solutions. The Complexity Break-Down Model (CBM) consists of essentially two distinctions. When evaluated
within the cycle time context, those distinctions crisply identify the key themes of CTR at Sun, the basic elements of the cycle time problem, and the solution approaches proposed by SPL. The top-level view of the model appears in Figure 6.

Looking internally first, one could characterize the left column of Figure 6 in terms of two key elements, "rules and tools." The rules are the key operating policies of the company, including how critical trade-offs that contribute to complexity are balanced. This is the domain of the policy makers and enforcers who set rules through behaviors, directives, and culturally based norms. For example, unchecked growth in the product portfolio has negative implications for inventory management and product forecasting. A rule that specifies, for example, an upper bound on the number of allowable configurations within a product family circumscribes the space of decision alternatives for a product marketer, and thereby provides a lever by which the organization can control or eliminate unnecessary complexity.

The rules or policies determine the nature of the tools employed. The term tools is used here to describe the resources that people draw upon in the execution of the business, such as processes, systems, and communication channels. Execution processes often arise as compensating mechanisms for ambiguous policy. The resulting processes waste time, require frequent intervention, and are needlessly complex. As they are reinforced over time, those processes become institutionalized as paradigms and perceptions about "how we do business." Note that paradigms are rarely assertions of value or principle, e.g., "we put our customers first" or "we don't compromise on quality." Usually they are ossified assumptions regarding procedures used to conduct business -- assumptions that frequently crumble when placed under the microscope.

As an example, at Sun we previously used a dynamic scheduling technique for scheduling orders. The technique assigned lead times to individual orders by applying a calculation that involved currently available supply, scheduled inflows to the factory, manufacturing throughput, and total outstanding backlog. The approach did not support predictable delivery because there were so many "free variables" used to determine lead times for individual orders.

One of the early successes of the SPL program was to dispense with the perception that we were stuck with the dynamic scheduling approach. Rather than treating lead time as a dependent variable that simply emerged from our complex soup of systems and order scheduling processes, we began to treat it as an independent variable or constraint around which to design systems and processes that ensured predictable delivery. We
effectively transformed lead time determination from an uncontrollable variable to a controllable variable. Or, in terms of the CBM's geographical layout, we "moved the issue north."

Figure 7 provides further examples of how the SPL team used the CBM to take inventory of factors that create complexity and inhibit fast cycle time performance. Figure 8 shows examples of some of the solution strategies the SPL team has developed using the CBM framework.

The usefulness of the CBM is twofold: to provide a rigorous assessment of cycle time issues and opportunities, and to flush out potential strategies for managing complexity. The constant quest is to transition those factors that are categorized as uncontrollable to the domain of controllability. Developing stable, adaptable processes based on a set of defined operating rules yields predictable results.

**Cycle Time Gravity**

CTR is, to a huge extent, about leverage. Analysis using the CBM helped SPL engineers identify and attack high-leverage pieces of the complexity pie to achieve cycle time breakthroughs.

**Figure 7: Using the CBM to Identify Inhibitors of CTR: What Problems Show Up Where?**

**Figure 8: Using the CBM to Chart Out CTR Strategies. One Key Approach: "Move the Uncontrollables North"**
quickly. Those breakthroughs help create what we call cycle time gravity\(^2\). The idea behind cycle time gravity is to instill behaviors both internally and externally that lead to a higher level of cycle time performance for an organization in a sustainable way. For Sun, CTR is not about reengineering, which typically concentrates on static, one-time redesigns. It is, rather, about changing mindsets, and then creating and continually refining methods that embody and implement those mindsets.

Sweeping reengineering plans are typically subjected to risk at the most critical juncture: implementation. Efforts to drive change can become watered down when large-scale programs are scattered across highly taxed organizations. The end result is low impact “incrementalism” that is frequently touted as continuous improvement.

Applying the concept of cycle time gravity to complexity management opportunities provides the greatest chance of success because it allows organizational participants to see, touch, and taste the benefits of CTR. The organization can begin to focus on high-impact change that can be incorporated into the process DNA of the company. Confidence in early CTR activities leads to positive feedback that propels and sustains larger CTR initiatives that require more fundamental shifts in organizational practices and mindsets.

**Conclusions**

Wetherbe (1995) has observed how cross-functioning can be used to achieve big leaps in CTR. In practice, it often appears to be one of the most difficult strategies of CTR to implement. The Decision Chain helps cross-functional participants see how their decisions shape or constrain the decision opportunities of other organizational participants, and thus provides a useful method for structuring cycle time analyses in large, decentralized corporations.

Cycle time reduction at Sun Microsystems is largely about identifying and eliminating unnecessary complexity. The Complexity Breakdown Model developed and applied by SPL has provided a rigorous but flexible framework for analyzing complexity and for charting strategies to reduce it. We believe these tools can help other organizations achieve cycle time breakthroughs by helping them to improve coordination and reduce complexity.

**References**


Tom Chavez, Ph.D., is a Systems Architect for Sun Microsystems, where he has worked on a number of projects in areas ranging from supply chain redesign and order fulfillment to product planning and adaptive network control. He completed his doctoral degree at Stanford in the Department of Engineering-Economic Systems and Operations Research in the area of decision analysis and probabilistic modeling, with applications to high technology product transition planning. Prior to Sun, he worked as a Senior Research Associate at Rockwell International and as a Marketing Director for Lumina Decision Systems, Inc.

Gregory R. Dion, is Director of Supply Chain Initiatives for Sun Microsystems Computer Corporation. Most recently, Mr. Dion has been responsible for implementing Sun's innovative product lead time and process cycle time reduction programs. Prior to joining Sun in 1992, he previously held a variety of operations positions at Wang Laboratories and the former Toy Group of General Mills, Inc.
Using Decision Engineering to Achieve Short Predictable Lead Time