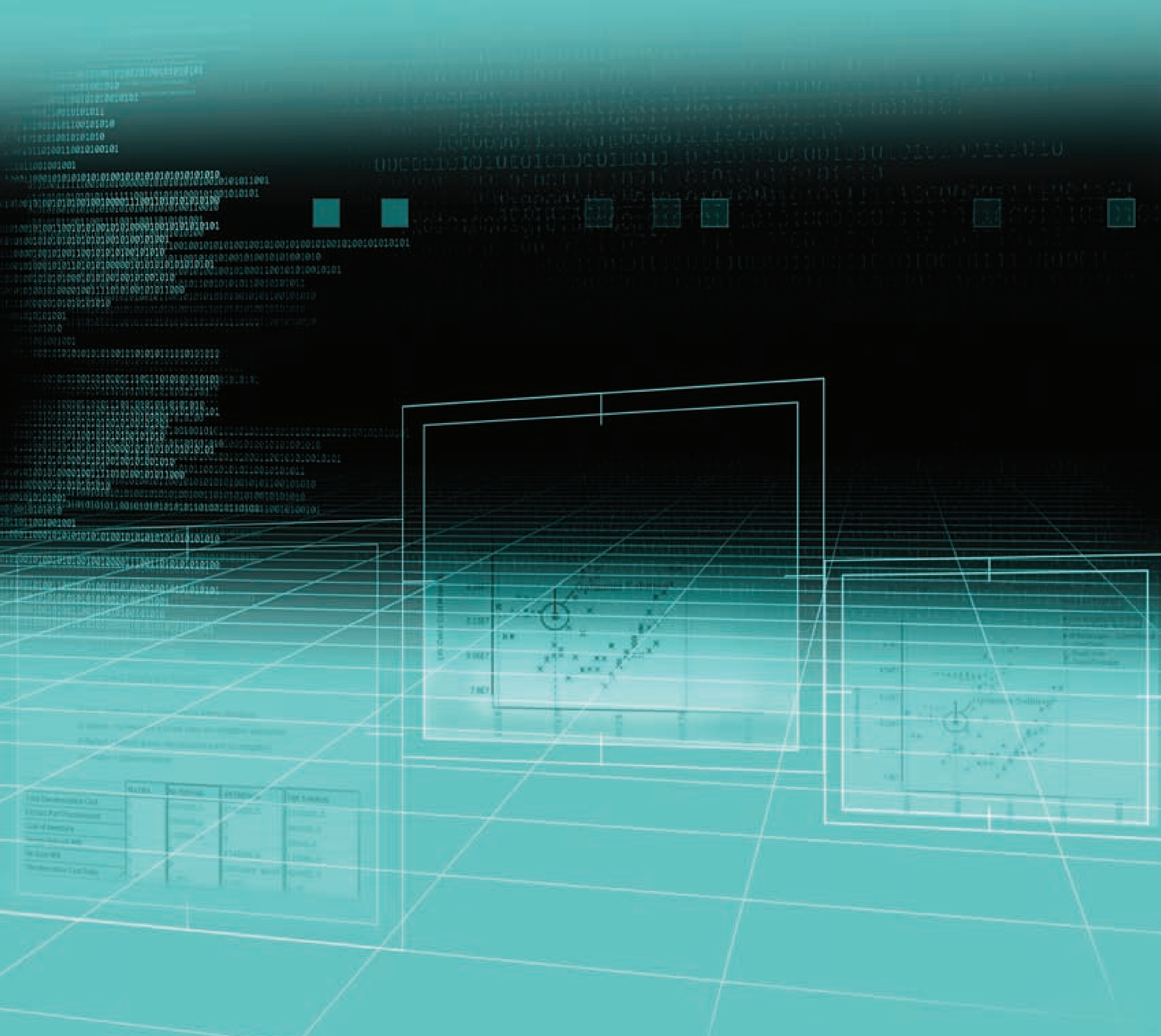
Strategic Management of DMSMS in Systems

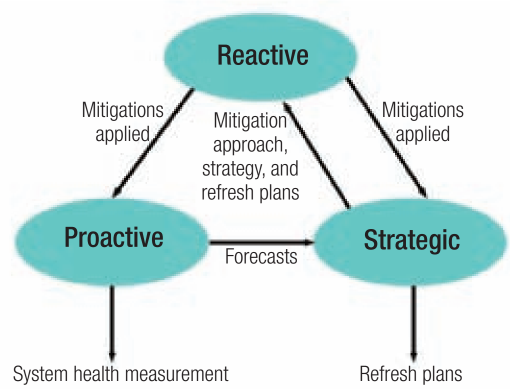
By Peter Sandborn



The escalating impact of Diminishing Manufacturing Sources and Material Shortages (DMSMS) on systems has resulted in the development of a growing number of methods, databases, and tools that address the obsolescence status of components, forecast future obsolescence risk, and provide DMSMS mitigation and management support. However, the majority of the existing offerings focus on reactive and, to a lesser degree, proactive management of DMSMS issues associated with electronic parts.

Effective long-term management of DMSMS in systems requires addressing the problem on three different management levels: reactive, proactive, and strategic. Figure 1 defines these levels and shows their interactions. To maximize the cost avoidance associated with managing systems, all three of the management areas should be considered concurrently.

FIGURE 1. DMSMS Management Strategy and Definitions



Reactive—When components become obsolete, determining an appropriate resolution to the problem, executing the resolution process, and documenting/tracking the actions taken.

Proactive—Determination of the status of the entire system with respect to DMSMS risk and assessment of the expected component needs against DMSMS risk, inventory, and spares status. Proactive requires an ability to forecast obsolescence risk for components. Proactive management also requires that there be a process for articulating, reviewing, and updating the system DMSMS status.

Strategic—Use system status, forecasted DMSMS risk, and expected needs, inventory, and spares to determine the mix of reactive mitigation approaches and design refresh (minor and major) that minimizes the life-cycle cost (i.e., maximizing cost avoidance) while continuing to meet all system requirements.

Strategic Management of DMSMS

Strategic management of DMSMS means using DMSMS data, logistics management inputs, technology forecasting, and business trending to enable strategic planning, life-cycle optimization, and long-term business case development for the support of systems.

Too often, programs become caught up in addressing obsolescence events as they occur, for example, making decisions on a case-by-case basis whether to undertake a lifetime buy of the obsolete part or to initiate a design refresh activity to replace the obsolete part with a newer part. This can lead to being caught in a "death by a thousand cuts" system management trap, spending valuable resources making a continuous stream of independent decisions about how to manage parts. Hindsight in this case often reveals that greater cost avoidance would have been realized by combining the management of many individual obsolescence events together into a single funded design refresh at a predefined date and bridge-buying sufficient parts to reach that refresh date when obsolescence occurs rather than trying to mitigate each individual problem to the end of the field life of the system.

This example is not meant to imply that the best DMSMS management approach for all systems is bridge-buy and refresh, but rather to point out that strategic management of DMSMS requires a broader view. It is not about making independent management decisions about each part in a list and then measuring results by accumulating individual DMSMS case-resolution metrics and cost-resolution factors. Strategic management requires the following:

- *A view that extends beyond individual electronic parts to boards, boxes, line replaceable units (LRUs), and so on.* Many things are not repaired, spared, upgraded, or replaced at the part (chip) level. Part-level obsolescence management is of little value to programs that never reach deeper into the system than individual circuit cards or boxes.
- *A view to all system components.* Obsolescence does not just affect hardware. Hardware and software obsolescence management must be coupled.[1]
- *A view to the enterprise.* Ideally, strategic solutions require coordination across multiple systems that share common parts and subsystems.
- *Applicable policies, technology upgrade plans, and other factors.* Such factors may constrain what DMSMS solutions can be applied, when they can be applied, and how they can be applied.
- *Decision making under uncertainty.* Everything that goes into determining a strategic solution is uncertain: obsolescence risks and dates are uncertain, resolution costs are uncertain, the end of support is uncertain. Finding optimal solutions that do not account for these and other uncertainties may be misleading.

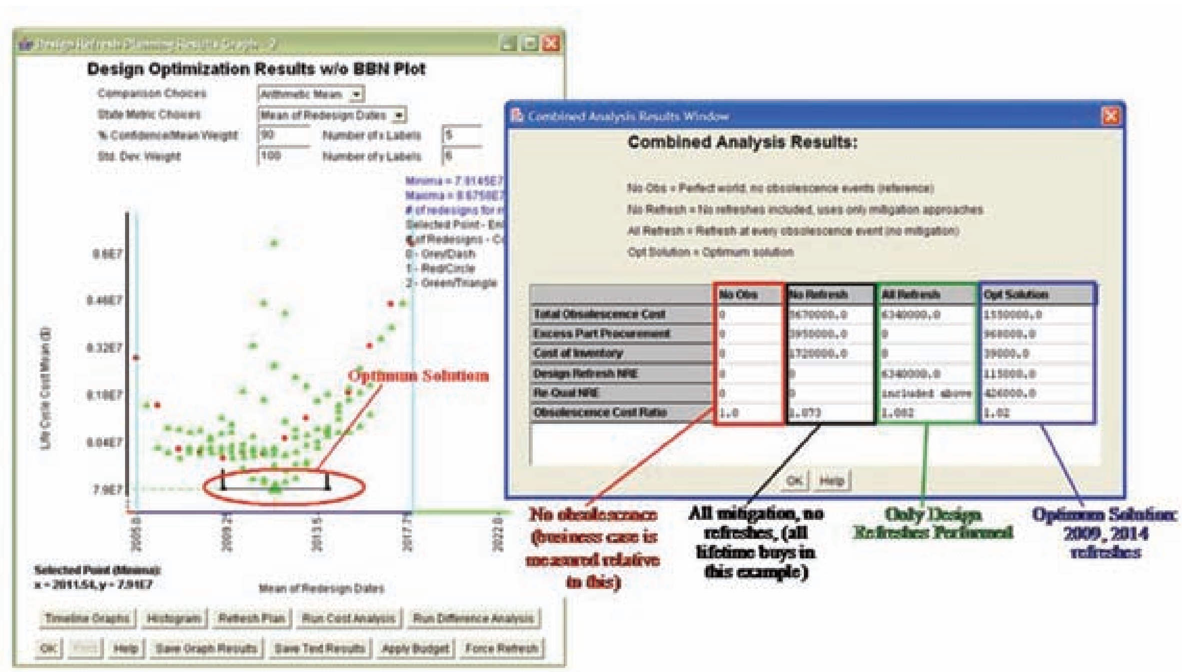
Building Business Cases to Support Strategic Management

Unfortunately, even when experienced DMSMS managers think strategically and propose solutions that have longer term impacts (e.g., planned design refreshes), they often cannot create the necessary business case support to convince the customer to take a strategic view.

A tool—Mitigation of Obsolescence Cost Analysis (MOCA)—has been developed to aid organizations in creating a plan for managing obsolescence and constructing associated business cases to support that plan. MOCA has been designed to generate a plan consisting of design refreshes mixed with reactive mitigation approaches where the total sustainment cost of the plan has been minimized.[2] MOCA takes as its input the bill of materials for a given system, along with the procurement cost and projected obsolescence dates or procurement lifetimes of the individual components (in this context, chips, circuit boards, LRUs of other kinds, or even software applications). MOCA can model multiple levels of hierarchy, so that an entire system or a system of systems containing common components may be loaded into the tool for concurrent analysis. MOCA also requires a production/deployment schedule as an input. This schedule may be supplemented with inventory status and a forecast of required spares. Using this information, MOCA creates a timeline of all possible design refresh dates that it couples with a timeline of all of the projected obsolescence dates for the components. MOCA generates candidate refresh plans consisting of zero refresh dates (all reactive mitigation), exactly one refresh date in the lifetime of the system, exactly two refresh dates, etc. The life-cycle cost of all the plans is computed, and the candidate plans are ranked according to the resulting life-cycle cost of the system.

Figure 2 shows an example output from MOCA. In the graph on the left side of the figure, each dot represents a unique refresh plan (the result in Figure 2 contains plans with exactly zero, one, or two refreshes in them). Corresponding to each plan, MOCA generates a list of components that are obsolete or about to go obsolete so that they can be refreshed. Parts that become obsolete before the designated refresh date are managed using a user-defined short-term mitigation scenario (in the example shown here, the parts are bridge-bought) until they can be replaced. The cost of the bridge-buy, along with the storage and handling costs and the costs of the design refresh itself (including

FIGURE 2. Sample MOCA Solution



nonrecurring engineering and requalification costs) are all included in MOCA's total life-cycle cost calculation for each refresh plan. The vertical axis on the graph is life-cycle cost, and the horizontal axis is time. The data points corresponding to the plans are plotted at the mean of the group of refresh dates they represent (one plan is expanded in the graph to show the actual two refresh dates it contains).

In order for the refresh planning predictions to be useful, the impact of the plans must be articulated as a business case. To evaluate the utility of the optimal plan, it is compared to a case in which no parts go obsolete, a purely reactive mitigation approach case, and a strategy in which every obsolescence event is resolved with a design refresh. These scenarios are compared by breaking down the total cost of obsolescence management into subcosts to identify where the money is being spent.

The true cost of obsolescence management can be determined for a given strategy by taking the total cost of the plan and subtracting from it the cost of managing the no-obsolescence scenario:

$$O_C = T_A - T_{LCP},$$

where O_C is the obsolescence management cost, T_A is the actual total life-cycle cost of the system with the selected obsolescence management approaches, and T_{LCP} is the total life-cycle cost in the no obsolescence scenario.

T_A includes all costs associated with procuring parts and building the system, all costs associated with design refresh and requalification costs, all costs associated with mitigation, and all inventory costs for storing parts. T_{LCP} includes only those costs that are not associated with obsolescence; it simply includes the recurring costs of building the system (if applicable) and procuring the parts. Thus, by subtracting T_{LCP} from T_A, the obsolescence management cost can be obtained.

MOCA breaks down the obsolescence management cost into the subcosts associated with the excess part procurement (the difference between part procurement costs if there was no obsolescence and part procurement costs associated with the mitigation of obsolete parts) as well as the inventory cost (cost of storing the parts over the long term). The obsolescence management cost also includes any costs associated with the redesign and requalification and any other costs associated with a design refresh. All the obsolescence management costs include cost of money (they are net present values indexed to the analysis starting year) and include the effects of the budgeting period duration. An example output from MOCA's business case analysis is shown on the right side of Figure 2 for a case in which all mitigation was either lifetime buys or bridge-buys.

Constraint-Driven Planning

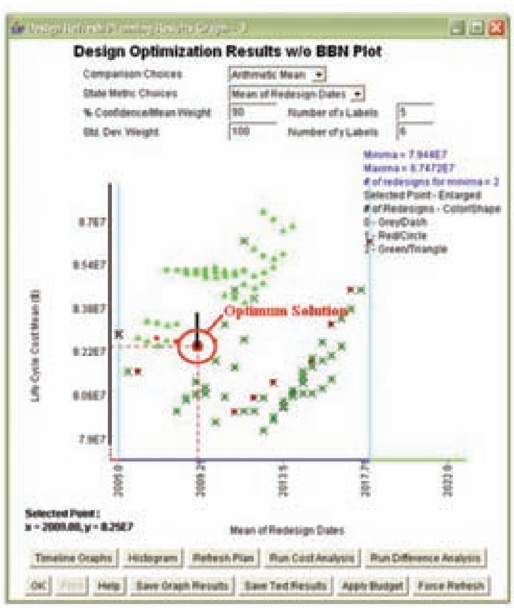
Constructing and costing combinations of mitigation approaches and candidate refresh plans constitute a significant step in the direction of strategic planning, but rarely is the management of a system this simple. Often, a plethora of constraints find their way into DMSMS management problems. The constraints may be budgetary (e.g., a ceiling exists on the expenditure that can be made on the system in a particular year), logistical (e.g., the platform is not available to be refreshed during a particular period of time or a finite throughput is associated with upgrading systems), or policy (e.g., a road map dictates that the system must be upgraded in a certain way during a certain period of time). In order to introduce constraints into the refresh planning process, the following obsolescence event types are used:

- *"Weak" obsolescence event.* No change to installed or new systems is required. As long as the obsolete item is available, new systems can be built and installed using it, and previously installed systems can be repaired with it if necessary.
- *"Strong A" obsolescence event.* Installed systems can continue to operate with the obsolete item until the obsolete item needs replacement due to a failure of the item. New systems cannot be built and installed with the obsolete item (whether the obsolete item is available or not).
- *"Strong B" obsolescence event.* Installed systems are not allowed to continue to operate with the obsolete item and must be backfitted within a defined time period. New systems cannot be built and installed with the obsolete item (whether the obsolete item is available or not).

As an example, Strong B events can be associated with the end of support of critical software components such as operating systems used in communications applications that connect through public networks. In this case, end of support means the end of security patches, after which the software represents a security risk if not replaced.

Figure 3 shows the MOCA simulation outputs after specific road-mapping constraints have been applied (the solution before constraints is shown on the left side of Figure 2). The refresh plans that do not satisfy the road-mapping constraints are crossed out in the graph

FIGURE 3. Sample MOCA Solution with Constraints Applied



in Figure 3. All the viable refresh plans (plans that satisfy the constraints) have been shifted upward in the graph because of the additional cost constraint that was applied to all design refresh plans with a design refresh between 2007 and 2010. The optimal refresh plan changes from a solution with two refresh dates (2009, 2014) to a solution with a single refresh date (2009) because of the constraint.

Closing Thoughts

Reactive management of DMSMS problems will always be necessary. However, strategic DMSMS management is possible and can lead to substantial cost avoidance for many systems. Use of strategic approaches such as refresh planning must be carefully tempered; in particular, when the required quantities of obsolete parts are relatively small, a careful analysis is required because, as so aptly stated by John Becker (former DMSMS program director for DSP), the "struggle to find duplicates, alternates or substitutes cost-effectively [creates] the illusion that some higher cost engineering solution or an end-product upgrade is financially attractive or the only option available" when it is not.

[1]P. Sandborn, "Software Obsolescence—Complicating the Part and Technology Obsolescence Management Problem," *IEEE Transactions on Components and Packaging Technologies*, Vol. 30, No. 4, pp. 886–888, December 2007.

[2]P. Singh and P. Sandborn, "Obsolescence Driven Design Refresh Planning for Sustainment-Dominated Systems," *The Engineering Economist*, Vol. 51, No. 2, pp. 115–139, April–June 2006.

About the Author

Peter Sandborn is a professor in the CALCE Electronic Products and Systems Center at the University of Maryland. Dr. Sandborn's group develops obsolescence forecasting algorithms, performs strategic design refresh planning, and lifetime buy quantity optimization. Dr. Sandborn is a member of the U.S. Navy TRENT Shareholder Council and is the author of the DoD DMSMS working group's DMSMS tool/data taxonomy. Dr. Sandborn also is an associate editor of *IEEE Transactions on Electronics Packaging Manufacturing* and a member of the editorial board of *International Journal of Performability Engineering*. He is the author of more than 100 technical publications and several books on electronic packaging and electronic systems cost analysis.