Event-Based Through-Life Cost Management

Robert A. Butler
TFD Group

Engineering life cycle cost models have traditionally been seen as inappropriate tools for both platform-level life cycle cost analysis and for budget-quality cost forecasting. The view has been that platforms (aircraft, ships, vehicles) are simply too complex to allow an adequate build-up of engineering data to support the bottom-up approach of engineering models during the early stages of acquisition. Moreover, the models have always been thought to be insufficiently rich (despite their hunger for data) to do an acceptable cost forecasting job for budget planning.

These perceptions began to change in 1996 when the Avondale Alliance made the first use of an engineering model, then called EDCAS 17 and now called MAAP[1], during the proposal process for development of the LPD 17 ship. Though not the favorite in the competition, Avondale won that contract and the U.S. Navy stated (and, upon a protest, the GAO[2] concurred) that the discriminating factor in the award was their proposal to use the model to manage through-life costs. Subsequently, MAAP was used as the source selection tool for some 35 (i.e., all of the) major systems selected for use on the LPD 17 ship.

More recently, the USAF, as part of a Total System Support Responsibility (TSSR) contract awarded to Northrop Grumman Corporation, has funded a strategic planning device called JCAPS (Joint Cost and Performance System) for the purpose of tying the budgeting process, not to historical projections, but to detailed modeling of the link between its near-term plans for the Joint STARS aircraft fleet and the resources required to carry out the plan. The heart of JCAPS is MAAP[3].

What makes MAAP different is event analysis. Basically, MAAP acquires its input data in the manner of a Logistic Support Analysis[4] (LSA) tool and uses the resulting data to generate cost forecasts. The essential difference between this and other engineering approaches to life cycle cost is that the unit of analysis is a failure mode translated to a maintenance event, rather than the traditional abstract construct of an average failure. By using the general data model of the LSA (parts are decomposed into events and events consume resources), MAAP achieves a

---

[1] Monterey Activity-based Analytical Platform. Trademark owned by Systems Exchange, Inc.

[2] The General Accounting Office, the analytical assessment arm of the U. S. Congress, used by government agencies as an expert honest broker. Subsequently, during the ship design process, the model was used for tender evaluation and source selection for 35 major systems used on the ship.

[3] A similar program, Total System Support Partnership or TSSP, has been approved and is going forward for the B-2 aircraft. The JCAPS system is being implemented under that program as well, with the addition of a Supply Chain Optimization and LSAR development functions. This expanded version of JCAPS will, more than likely, also be used on Global Hawk, another TSSR program and is under consideration for use in the JSF program’s Autonomic Logistics function.

[4] This term was first introduced with the publication, in 1973, of the first LSA standard, MIL-STD-1388. The intent of the new form of analysis described in that document was to introduce logistic discipline into the system design process. It was produced during the same period that the author was starting to refine his own ideas about introducing life cycle cost discipline into the design process. The standard was largely ignored in its original form. The later publication (1981), under Russell Shorey’s sponsorship, of the -1 version, broken into a task structure in Volume A and a data delivery schedule in Volume B, produced a veritable world-wide revolution in the use of logistic analysis. Both this, and the 1991 update, MIL-STD-1388-2B were subsequently abandoned by the US Department of defense in the mid-90’s as excessively expensive and unproductive. They and derivative standards continue in use, however, both in the US and elsewhere, the most important of these being DEFSTAN 00-60.

level of detail and accuracy in cost forecasting not previously seen. By demonstrating the feasibility of acquiring, organizing and storing the requisite data rapidly, MAAP has also proved itself a practical tool.

The following discussion explores the pros and cons of various cost analytic methods, what makes them attractive or unattractive in various usages and why the method exemplified by MAAP, while infeasible in the past, appears to be the best method to explore whole-of-life cost issues in the current era. We beg the reader's indulgence regarding the frequent references to MAAP – it is a platform for research into the cost-analytic propositions presented in this paper, and a work in progress. It is worthwhile to report this progress as an illustration of the feasibility of the approach.

The Uses of Cost Analysis

The first responsibility of a cost analyst is to determine to what use his work will be applied. Invariably, the cost estimate is used as support for some form of decision. It is the *type* of decision that concerns us, however. There seem to be two major classes of decisions to which cost is applied: a) which option should I choose? and b) should I choose any of them? [5] While these might at first seem quite similar, not only are they different in character, their impact on methodology can be profound.

The first type of decision might be thought of as a resource allocation decision. We compare one thing to another and decide which we want to pursue on the basis of many factors, one of which is resource cost (another, of course, is benefit). The cost-versus-benefit view can be applied to very complex "baskets of goods" or alternative courses of action. The objective of such analysis is to determine which alternative is the best to choose – if we choose to do anything. A significant aspect of this kind of decision is that it does not demand great precision in the results. It is enough to know that option A will be less costly than option B. Knowing how much either one will really cost is not only more difficult, some might argue that it is, in reality, impossible.

The second type of decision is usually thought of as an affordability assessment. It seeks to compare the quantity of resources required to carry out an alternative with the quantity available. These cost concepts – how much we need and how much we have – require a great deal more precision. The daily news for most of my career as a cost analyst has been filled with scandalous over-runs and the like. Equanimity in the face of these gross failures of cost analysis can only be explained by the fact that one worked in the "which option" world rather than the "how much" world.

The vast majority of people who pursue cost analysis professionally take the seemingly reasonable view that the purpose of analysis is the second of these – the affordability assessment or "how much" type of analysis. Generally speaking, we find them in the employ of those involved in acquiring, distributing and managing budgets. Their concerns are accuracy, time and budget category separation of their estimates, rather than the reasons why the estimates are of a specific value. Generally, they have little interest in costs that extend for more than a short time span because the business of budget management rarely extends beyond a relatively few years.

There is an entirely separate group of analysts, however, who have generally labored in the arena of life cycle cost[6] from the point of view of the engineering community. Their concern has

[5] Another way to distinguish these two decision types is a) what to do and b) how much money is required to do it.

[6] New-comers to cost analysis may find the names of cost concepts confusing. Herewith a brief modern history: the first term was ownership cost to distinguish between the act of buying and that of owning a system. This term was in

been more aligned with the first of these motives, specifically, determining which among alternative designs, vendors, support policies, test regimes, training programs and so on, would lead to the most advantageous trade-off between cost and benefit. These people generally function as advisors to design teams or system buyers.

Neither of these groups or approaches is correct or incorrect: they are simply different. What makes the difference interesting from our point of view is that the alternative uses of analysis lead to different types of analytical tools. Basically, those interested in affordability have used whatever technique they could find that would most reliably predict how much. Those interested in resource allocation decisions are constrained to use cost methods that not only tell them how much, but why. Since the *why* is the real objective, the *how much* is only needed to distinguish between alternatives. The emphasis on accuracy is correspondingly less.

It should be clear that through-life cost management requires support for both kinds of decisions. The *how much* methods are always necessary to secure the funds required. But cost-effectiveness of the pursuit, whatever it is, can only be enhanced by methods that help us select the best course of action, the *what to do* methods.

Alternative Cost Analytic Methods

The most common method used by financial cost analysts is called parametric cost estimation. While the methods can be extremely sophisticated[7], the essence of the approach is the use of statistical analysis methods to relate cost outcomes to measurable (and more readily predictable) variables. To the consumer of the cost results, these methods often appear to be simple. The final result of the research may be a few equations requiring that a few variables be estimated and input, producing an answer almost instantly. Unfortunately, it may take several months and more than a few bags of gold to do the research necessary to estimate the parameters of the cost equations. Even more unfortunately, there is a problem with the persistence of applicability of the parameter estimates – the world changes and invalidates the parameters as a result.

Financial cost analysts also use a less sophisticated (and possibly even more ubiquitous) instrument for cost forecasting – the spreadsheet. A marvelous invention, the spreadsheet places in all our hands the means by which to build complex analytical castles whose structural integrity is akin to a that of a grass shack. The most sensible of these are instruments are charts of accounts (cost breakdown structures) and little more. Behind each cost element there is usually some kind of rationale or equation that explains the number found in that cell. Most of the work in performing this kind of analysis is the research and calculation needed to produce each of the numbers. The advantage of the spreadsheet analysis is that it is easy to build and understand. The disadvantage is that it responds poorly to changes or the need for “what if”

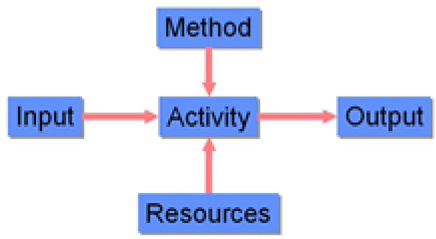
---

vogue in the late 1960s and early 1970s. By 1975, the profession had shifted to the term life cycle cost. [I would love to describe reasons for these changes, but have never discovered any fit for print.] Largely because few were interested in it, life cycle cost or LCC persisted for a very long time. EDCAS-17, born in 1996 was a life cycle cost model. By the time its name was changed to MAAP in 1997, it had become a total ownership cost (TOC) model. [Had the first study been done for the USAF, this would have been total cost of ownership or TCO. USAF have apparently abandoned this nuance and allowed the Navy to lead in the terminology war.] Meanwhile, the Australians and British were using the term through life cost (TLC). While the former persist in that, the British, more mobile in such things, have moved on the whole of life costs (WLC). The thing for the newcomer to remember is that there is *no difference whatever* in these concepts, correctly implemented.

[7] The author is quick to admit a fairly complete level of ignorance about these methods. The few times he has found himself in proximity with those who ply this trade, both he and they have regarded each other much as a Martian might look at an earthling.

analysis. When the fleet size, for example, changes from 25 to 50, some of the elements may actually double, but most won't and they'll have to be recomputed from scratch.

A new method recently introduced in the affordability or budget-driven cost analytic world is activity-based costing or ABC. This method is ostensibly a response to the complaint that budget estimation based on historical tendency failed to account for intended (and known) changes in operations. The consequence is a method that carefully traces the relationships, in a given enterprise, between the input to a production process, resources required to carry it out and the anticipated output. These elements are illustrated in Figure 1.



1. *Figure 1: Activity-Based Costing Logic*

If the method is planned to remain the same, then a planned increase in output will lead to an increase in input resources, which forms the basis of the next interval's budget forecast. This method has been greeted with enthusiasm because it infuses a bit of reality into the budget forecasting process. As time goes by, however, some weaknesses in the method have become evident. While not as costly as the historical research required to develop trustworthy parametrics, ABC nevertheless claims large amounts of research resources and, like parametrics, finds that the results are invalidated all too quickly by changes in methods or in the character or scale of operations.

The other group of analysts – those interested in resource allocation decisions – have, until recently, used radically different tools. These methods have a variety of names, including engineering cost methods, bottom-up methods and accounting models. I prefer the term process models to indicate that, rather than finely drawn reduced-form equation sets, these models are over-specified equation sets that use separate equations to gather functional costs into discernable groups[8]. The term process model was first used in conjunction with models whose purpose was to trace the cost-creating logistic processes that arose from interaction between the attributes of hardware elements (MTBF, MTTR, unit price and so on) and the operating and support regime to which the system of which they were a part was subjected.

An example of a process model equation is:

$$Cost_{Training} = \sum_{j=1}^{m} CC_j \sum_{i=1}^{n} B_i \left(1 + ToR \cdot LC\right),$$

in which CC_j is the cost to train a single person on the j^{th} of m system components, B_i is the integer number of positions (billets) requiring training at the i^{th} of n bases, ToR is the job turnover rate or the portion of B who will have to be replaced each year and LC is the number of

[8] I trust the reader will forgive this jargon. A reduced-form cost equation set reduces all the instances of any independent variable to the smallest number necessary to compute the single value, cost. An accounting model allows the re-use of the variable over and over again as it might influence different cost elements and is, thus, over specified.

years in the life cycle. The process modeled by this equation is that of filling all the trained positions initially (the 1 in parentheses) and then keeping them filled through the course of the life cycle. The prior calculation of *B* would respond to how heavy a maintenance or operational burden was imposed by the operating pace and the frequency and duration of maintenance events, while the cost, *CC*, would respond to the complexity of the tasks to be performed by people occupying the job positions.

Note that in that simple equation, the cost of training responds in different ways and to different degrees to:

1. hardware design (the length and therefore the cost of the training course and the frequency, duration and variety of maintenance events)
2. deployment (number of bases)
3. operating pace (hours of operation related to operating program and numbers of systems at each base, which interact with the frequency of maintenance and operating events, causing their numbers to change)
4. support structure and policy (location, number and types of maintenance carried out at bases at different echelons, personnel characteristics such as job turnover rate of trained people[9], policy for repair, discard and use of scheduled maintenance)

These models – simple linear combinations in most cases – are very familiar to members of the logistics community, but not to the cost analysis community. Even though they are used to estimate costs over the life of a system, only naïve program managers have ever tried to use them for budget forecasts. They have been considered, and rightly so, too inaccurate to be used in financial planning. They *are*, however, useful in distinguishing between different designs or vendor offerings on the basis of the alternative designs' relative contributions to life cycle costs.

During most of the history of life cycle cost analysis, there has been a hard-and-fast separation between the use of process models in design and logistic planning and the use of other methods for financial planning. This situation is not ideal. Almost any program manager would prefer that the same instrument be used to guide the decision process of determining what to do and determining how much money is needed to do it.

That distinction has begun to break down with the introduction of MAAP (there must also be other methods of a similar nature being used here and there), together with a growing dissatisfaction with the inability of traditional financial forecasting methods to deal with significant changes in operational pace and mission.

The ability of a process model to react quickly to what-if questions based on hardware or operating and support environment changes is most desirable. The accuracy anticipated from parametric or ABC methods is also crucial. None of these methods has been capable of producing both desirable results simultaneously. MAAP offers the promise of doing so.

[9] The job turnover rate and, more important, the personnel turnover rate (the later meaning losses to other employers rather than transfers to other duties) will frequently respond to what economists refer to as investments in human capital. That is, training people enhances their market value, both to their current employer and to all other potential employers. Investing in such capital always carries with it the risk that this enhancement will lead to a decrease in retention. The US Navy experienced this problem in an acute form during the 70s and 80s and had to introduce a program of selective reenlistment bonuses to counteract the effect.

Event-Based Through Life Cost Analysis

Most of the labor of a cost analysis is expended collecting, checking, collating and reformatting data. This explains why most cost models built in recent years have tried to limit the amount of input they require. However, if, somehow, the amount of work required to marshal the input data were drastically reduced, it would also be true that a model capable of using more and more input data would, more than likely, be capable of producing more accurate and comprehensive answers. Thus, if one had more data, it would be convenient if there were a place for it in the model.[10]

The ideal situation is a modeling process that is tolerant of limited information so that it can be used at the outset of a decision process, but can also make use of the virtually unlimited amounts of data that become available as systems mature. MAAP was designed to respond to the most abstractly simple representation of systems, as well as the most realistically complex.

A consequence of the now-past era of MIL-STD-1388 is that enormous data sets are available on the majority of in-service systems in the US and large numbers of systems in other countries, all of which are sorely underutilized. MAAP's ability to utilize almost the entire range of inputs, in any level of detail, found in a standard LSAR data collection is one of its more interesting features. Because the model allows the user to alter any input value, from hardware descriptive elements to basing, operations and support regime, quickly and easily – and then re-translate those changes into a time-based resource requirement and location stream – it breaths new life into these often disused data collections.

The basic approach of event-based analysis is straightforward. The life of a system (or systems) can be thought of as a collection of events, each of which occurs with some frequency, takes a certain amount of time and claims certain resources, either consuming them entirely, using up a portion of their life or merely dominating them for a period. Events of different types are related to each other as illustrated in Figure 2.

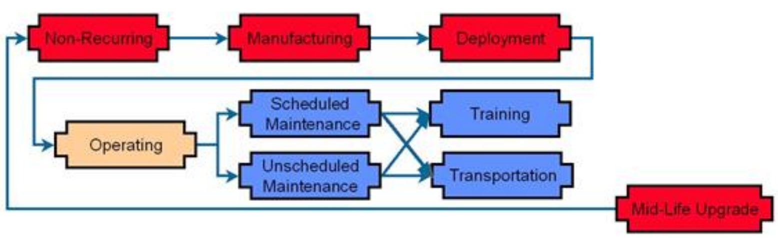


Figure 2: The Event Chain

[10] It is, perhaps, disingenuous to say this. The bigger issue, hinted at in the first sentence of this paragraph is data management itself. The practical utility of a model that uses vast amounts of data is limited by the corresponding utility of its database and the user interface that allows the modeler to gain access to the data he needs. MAAP was implemented on the TFD Database (TFDdB) rather than on one of the products derived from MIL-STD-1388. As a consequence, data organization, data element definitions, table definitions and relationships all lend themselves to a) use in mathematical formulations without further transformation, b) ease of accurate re-use when surrounding conditions have changed and c) flexible re-definition or re-aggregation into forms required by competing standards for data delivery. A further attribute is in process of being added to the TFDdB for the B-2 JCAPS implementation in which supply chain consideration make it desirable to feed the real-time transaction stream (of data) to the asset manager. Because the transaction stream passes through the TFDdB on its way to TAMS (the Tactical Asset Manager's System) it can also be analyzed and captured in a data warehouse where, periodically, we apply demand and delay forecasting tools to provide both automatic updates of the static data collection (both mean and variance), but also provide performance measurements for re-suppliers and repair vendors (including trend analysis for delay times).

Each type of event leads to the next, from non-recurring research and development, test and evaluation activities, through manufacturing, fielding (installation and check-out), operation and resulting maintenance. The entire cycle starts over again for a part of the system at the event of a mid-life upgrade. This is especially important for systems in continuous upgrade as is common with modern systems based on off-the-shelf technologies.[11] Newer systems, in fact appear to run through this entire cycle over and over again, each cycle actually initiating long before previous cycles have worked their way through the fleet.

An event itself is a microcosm of cost analysis. Figure 3 depicts the data model for a MAAP event, which consists of component-centric data tables and relationships. Each box represents a table in the database with its own key structure and attributes that match the keys. The lines represent relationships (key inheritance) that make it possible to recover “downstream” data – data sharing the base key but containing other keys as well – knowing only the identity of the part and its location in a system.

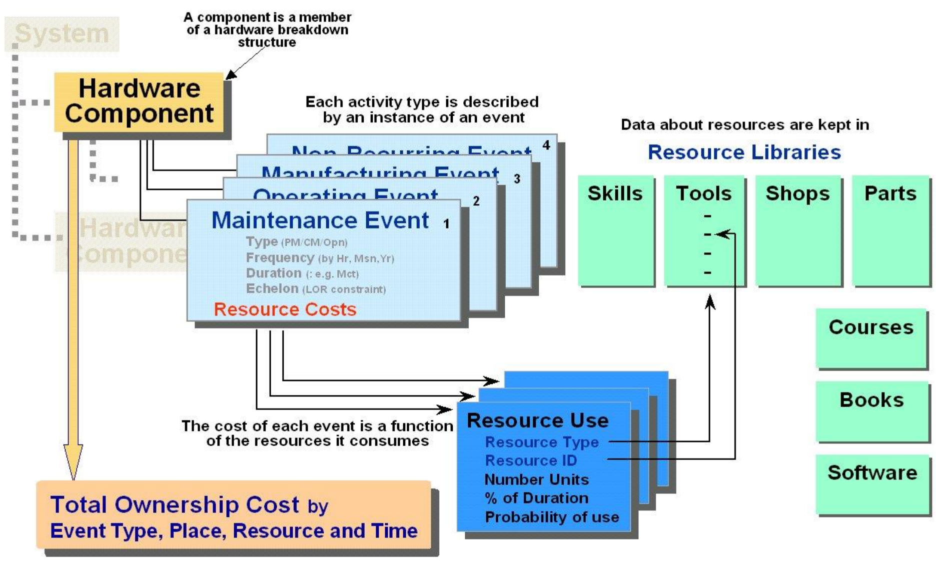


Figure 3: Event Data Model for a Component of a System

As you study the graphic, notice that a single component of a system, which may have thousands of components, can be associated with many events and each event with many resources of each type, and that every resource may have its own similar collection of data. It is not surprising to consider that such a vast data collection provides the underpinning for cost estimation of “budget quality.” Don’t forget, however, that by using abstract aggregates for these concepts (for example, the component could be an entire ship, factory, vehicle or aircraft) the inputs could be simplified to an extreme level, enabling the analyst to perform rapid system or fleet level analysis using exactly the same basic concepts.

---

11 Continuous improvement becomes a necessity as the components used to make a system go out of production as quickly as every 18 months. The notion of a life-time buy of spares has long been discredited as wasteful and, usually, futile because of the uncertainty of demand predictions. An interesting sidelight to this phenomenon occurred during the design of a fuel system component proposed for the F-35 (Joint Strike Fighter). While working as LCC consultants using MAAP to develop cost forecasts for this module, the system designer made it clear to us that the system that would actually be built would bear very little resemblance to the one proposed. Everyone involved on both sides of the contracting fence understood that at least *two* generations of technology would be superceded before the actual design destined to be built (one imagines in only the first block of fighters) could be completed.

At the upper left of Figure 3, the faded elements represent a system hardware breakdown structure of which the hardware component depicted is a part at any level in the structure. The box labeled Hardware Component has data that relate to the part itself such as its name, part number, weight, volume and so on. A collection of events of all types can be associated with a given part, each one containing information about the type of event, its frequency measurement, duration, echelon or location of performance and resource associations. Each resource association links a resource type (part, tool etc.) to the maintenance event, which is in turn associated with the hardware component.

The volume of data in some of these entities is small – an event is restricted to frequency, duration, echelon and a few other special purpose values, and a resource use row is even simpler. On the other hand, the resource libraries contain descriptions of items similar in some respects to the component itself – for example, a second-order model is capable of modeling the maintenance regime of the support equipment used to repair the main component. Altogether, a staggering amount of data might actually be attached to the computation of life cycle cost for a single component (and every other component) in a system.

When all these elements are combined, the system itself is deployed and operated, causing operating time or cycles to be logged to the component. These measures (as well as the passage of calendar time) accumulate for every instance of the component, ultimately tripping the frequency measure of each event type. The systems were deployed geographically, so the computation knows both the time period and geographic location of the resulting demand for resources. The time required to perform maintenance events is taken away from the potential of fleets to perform assigned missions, which in turn leads to a reduction in measured operational availability.

The equation system of MAAP was deliberately constructed to be as flexible as possible in accommodating different customer requirements for cost accounting frameworks or charts of accounts. This “flexible chart of accounts’ is accomplished by the use of the concept of a “cost atom,” a measured cost element (i.e., equation) so small as to be of little direct interest to any potential user. The MAAP model, as currently constituted, contains (depending on how it is run) between 2,000 and 10,000 such cost atoms. The pay-off of such detail is that virtually any cost breakdown structure defined by an end-user can be replicated by simply re-assigning cost atoms to a new hierarchical collection of “cost molecules.”

The detail of the inputs provides for the desired realism and accuracy in budgetary processes. The detail in the outputs affords flexibility in meeting end-users’ requirements for visibility of their often unique cost accounting elements.

From Life Cycle Cost to Budget Management

Early in the conceptual development of the JCAPS system, it became clear that, while life cycle cost forecasting was interesting and useful, there was a far greater pay-off related to the potential for bringing rigor to the budget forecasting process. Doing so placed two demands on the modeling process. The first was simply an accounting matter – insuring that the time-slices for which cost estimates were made aligned properly with the budget and expenditure reporting cycles. The second requirement is far more demanding.

Perhaps the most demanding (and unfortunately most common) budget-related task for a manager is to figure out how best to cope with a budget cut. The ramifications are both short- and long-run. The former are normally easy to gauge, but the latter usually result in unconvincing arm waving. The virtue of a life cycle cost model is that it understands the long run. The virtue of a life cycle cost model that re-computes resource quantity requirements in each time slice is that it gives visibility to the quite irregular ups and downs of costs that respond to operational and maintenance irregularities. For example, accelerated operation during period t_1, at the sacrifice of some delayed maintenance, will inevitably lead to a compounding effect in t_2 or later from a) the maintenance that would have come due any way in t_2, b) the increased maintenance resulting from the increased operation in t_1, and c) the maintenance deferred from t_1, but not further deferrable.

But the ability to simply trace out the time period impacts of such changes is not quite enough. Because we are dealing with a model, there is also the possibility of introducing optimization into the process. It was decided that this should also be done with the MAAP version that found its way into JCAPS[12]. The forthcoming MAAP release contains an optimization algorithm that allows both forward and backward movement along an optimal expenditure curve that relates incremental investments in all classes of resources (parts, tools, people etc.) to achieved operational availability. This "resources to readiness" mapping was among the original goals for the model that gave rise, among other things, to the choice of its name.

The optimization process called mBOSS (MAAP Budget Optimized System Support) is a member of the class of models called marginal analysis models, the first of which, METRIC, was developed by Craig Sherbrooke at RAND in 1966[13]. The METRIC model and its successors, all the way to the current generation called VMetric[14], are spares optimization models. That is, they look only at the single resource class, spare parts. mBOSS represents a generalization of that concept to include all the resource classes required to design, build, operate and maintain complex systems.

Appreciation of the way a marginal analysis model works is crucial to understanding the mBOSS algorithm. In the simple illustration in Figure 4, a system is modeled as a collection of removable assemblies called LRUs (line replaceable units – components that can be removed and replaced directly on a system to restore it to operation). Each of these will be considered for inclusion in an inventory of spare parts.

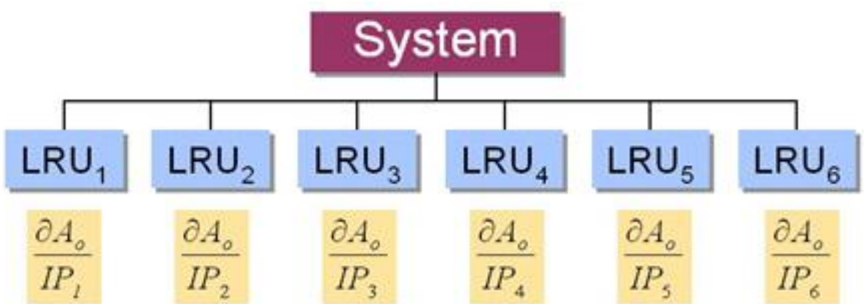


Figure 4: Calculations for Marginal Analysis

---

12 I would like to acknowledge Richard Gunkel's profound impact on the design of the current version of MAAP. Gunkel played a significant role in convincing the USAF to accept the TSSR idea and in the process was able to convince senior decision-makers that the MAAP content of JCAPS could revolutionize a budgetary process that had become quite evidently ill. This emphasis was then translated into the technical requirements of MAAP whose fulfillment is reported in this section.

13 Sherbrooke, Craig, "Multi-Echelon Recoverable Item Control," RAND, Santa Monica, 1966.

14 VMetric® is the name of a line of spares optimization models based on Sherbrooke's VARI-METRIC work, whose calculation engines were developed by Sherbrooke for TFD Group. The trademark VMetric® is also owned by TFD Group.

The ratio shown underneath each LRU compares the change in availability, ∂A_o, (or any one of a number of other measures of effectiveness[15]) resulting from increasing the stock of that LRU by one, to the unit price, *IP*, of that LRU. The price remains constant, but the increase in availability diminishes as the stockpile of that LRU is increased by each subsequent purchase. To determine the most economical combination of parts, we use marginal analysis to do the following:

1. Choose the LRU with the highest ratio
2. Re-compute the ratio for that LRU
3. Repeat steps 1 and 2 until the target A_o as reached, or the budget exhausted

The target maybe an availability target or we may be looking for the highest availability we can achieve for a fixed budget or budget increment. The algorithm stops when the target or budget limit has been reached.

Figure 5 shows the typical shape of an mBOSS optimization run. As one would expect of any phenomenon related to a production function, investment at first has a high return in

Figure 5: Multi-Resource Availability-for-Investment Curve From mBOSS

terms of increased availability, but diminishes as more and more investment is made. Ultimately, increased acquisition of resources has little or no effect on the availability achieved, which is usually constrained at the top end by the inherent properties of the system such as its reliability and maintainability measures.

Figure 5: Multi-Resource Availability-for-Investment Curve From mBOSS

---

[15] A staggering variety of such measures is in use in various industries and agencies. These include, in addition to operational availability, mission capable rate, fill rate, backorder rate, issue effectiveness, confidence level of no stock out, on time departure rate (airlines) and inter-arrival rate (transit systems). Most of these rates are mathematical transforms of each other, but they do not necessarily behave the same way – that is, optimizing against one measure will give a different sequence of choices than against a different measure.

The most useful part of the marginal analysis is that it leaves behind it a trace of how any given level of effectiveness has been reached. That is, the calculation results in a list of the sequential choices of each additional resource and the location at which it must be present to support the event activity at that time and place.

Every entry in the list, a small section of which is shown in Figure 6, indicates the sequence in which it was chosen, its marginal influence on cost and availability and the events that determine the need for these increments in resources. This list and the numeric values of availability and cumulative cost included in it are represented by the dots from which Figure 5 was constructed.

Year	ResourceName	UnitName	Delta	Cost	DeltaPerDollar	RunningCost	Ao
2001	Assembly E4 type 1	Operating Unit 01	0.0069916613	44948.75	1.555474E-07	333618124.09	0.6927041456
2001	Assembly E6 type 2	Operating Unit 03	0.0496814467	319805.59375	1.553489E-07	333937929.68	0.6931743280
2001	Assembly E4 type2	Operating Unit 03	0.0464487146	299722.09375	1.549726E-07	334237651.77	0.6936204825
2001	Assembly E3 type 3	Operating Unit 03	0.0488565757	316171.4375	1.545256E-07	334553823.21	0.6940473056
2001	Assembly E4 type2	Operating Unit 01	0.0463056126	299722.09375	1.544952E-07	334853545.30	0.694369629
2001	Assembly E5 type 2	Operating Unit 03	0.0560656698	363415.4375	1.542743E-07	335216960.74	0.6949259358

Figure 6: Multi-Resource Investment Sequence Table From mBOSS

Notice in the table that there are not only several resources of several classes (called types in the ResourceName column), but that they are also distributed to different locations. Thus, mBOSS is providing, not only optimal quantities to meet system targets, but also optimal geographic distribution of those quantities, according to the geographic distribution of demands. The algorithm does this for every time interval desired, such as a year. The end result is to translate operational plans (basing, scenario, operational pace and profile) into optimal resource requirements by time and geographic location.

It is useful to digress for a moment to compare the capability provided by this type of computational procedure to that anticipated by the authors of the MIL-STD-1388 family of standards. There is a significant increase in the utility of resource requirement estimation when moving from the relatively simple world of LSAR generation and through-life modeling based on the data. Table 1 compares the bare LSA data collection to the outputs available by processing the data through MAAP.

Dimension	MIL-STD-1388-2A/B	MAAP
Resource quantity	mean requirement	optimal requirement for A_o target
Time	program maturity	each time period (month, quarter, year…)
Place	echelon of maintenance	actual geographic place

Table 1: Utility Contrast Between LSAR and MAAP Data

The mean resource requirement is nothing more than a starting point to understand what resources will actually be required to deliver a given system effectiveness. Knowledge of requirements, even if computed correctly, at program maturity is of little value for programs like Eurofighter and Joint Strike Fighter, both of which may be in production for longer than the useful life of the first blocks. Requirements by echelon are similarly of little value when most organizations have barely discernable echelons – they are, in fact, abstract constructs meant to make logistic systems more understandable.

Returning to our main theme, if we have determined the best way to spend a quantity of money in a given budget period, as represented by the curve and table above, then we can also discern the most advantageous choices of what to surrender when confronted by the need to cut the budget. That is, we simply walk backwards down the table until we've "given back" the amount of the cut. Because the table was constructed in the first place by only adding each resource as it proves to be the next best (i.e., less valuable than the previous addition), walking backward always assures that the next resource to be divested will cause the smallest reduction in availability.

Note that the process does two things not actually done before in a rigorous way. First, it helps choose the *least valuable resources* to divest and second, it determines *the amount of sacrifice in availability* resulting from the cut. As mentioned above, availability is only one of several related measures that could be manipulated by such a computation. Thus, the method not only maps resources to readiness, but does so in the context of policy advice regarding the best (most cost-effective) response to budget changes.

Conclusions

The concept of life cycle cost has been with us for the better part of 50 years. It has been relegated to a relatively minor role until recently, for a variety of reasons, not least of which was the perceived inaccuracy of its results. In more recent times, as public budgets shrink and costs rise, interest in life cycle cost has risen to a quite serious level. The lucky admixture of extreme computational capacity widely available on everyman's desk top has also made it possible to devise ways in which the traditional shortcomings of analytical life cycle cost models can be overcome. MAAP is an example[16].

By tracing the cost consequences of inherent hardware attributes as they are influenced by the operating and support environment in which systems are deployed, we can achieve an accurate, agile estimation of cost.

Accuracy grows from the level of detail in the input detail and the extent to which the data are manipulated by realistic process algorithms. The single-equation training model cited above, for example, is replaced in MAAP by an algorithm that requires some 35 pages and 60 equations to explain[17]. The difference in the estimate of cost is not necessarily very large – but it would seem quite large if you had to carry out the training and had underestimated your budget requirement by that amount. The use of such extreme complexity would never have been contemplated even 10 years ago, no less in the 70's, before the first personal computer was produced by Altair.

Agility is a reference to the model's capability to produce new answer sets in response to changes in problem definition. The essence of this attribute is the combination of ease-of-use represented by the form of the user interface and availability of the crucial variables for manipulation. It does no good to make changing inputs easy if only a sub-set of inputs can be

[16] MAAP is in use by a small, but active and powerful community. It has, at this writing, nearly completed a lengthy cycle of verification and validation by the UK MOD as the only approved tool for whole of life costing. It is in wide and rapidly increasing use at both RAF Wyton and BAE SYSTEMS in the UK. Several programs at Northrop Grumman have adopted the use of the model, including Joint STARS, B-2, Global Hawk, LPD 17, MP-RTIP and others. At Lockheed Martin, the Joint Strike Fighter group has begun using MAAP, and the Multi-mission Maritime Aircraft are preparing to use it as well. Other users like Parker-Hannifin have been using MAAP for some time, not as system integrators, but as component suppliers motivated by the need to understand how they might best organize themselves and their activities for performance-based logistic contracting. The model has also been used on behalf of the US Coast Guard (Deep Water), the US Navy (SURTASS) and a number of firms involved in proposals for various programs.

[17] It is difficult to illustrate the sheer bulk of computations in MAAP. Consider, for example, a relatively simple problem involving a 10-year life cycle and 2,000 maintenance events. MAAP catalogues the number of each event that occurs on each *day* over that period, calling for more than 7.3 million combinations to be explored. This is only the first step. Next, demands for each resource type in seven different classes of resources (there are normally thousands of resource types within the classes) must be posted for each day. These are then accrued to the appropriate time intervals. The granularity of days is required to be able to accommodate variable time-slices.

changed in that way. The prejudices of the model-builder must be held in check as he goes about deciding what to make available to the user and skill is necessary in formulating the method of presentation of those variables to insure that the user will make proper use of them in constructing his scenarios.

We have come a long way, technologically, in the past 20 years or so. Database technology has made it possible to organize and manage staggering amounts of data. Clever database design can also minimize the extent to which the data must be manipulated for each new problem. Computational power and memory capacity have grown to such a degree that calculations can now be done routinely that simply couldn't have been contemplated as recently as ten or even five years ago. The general user interface available to personal computer users has done much to make virtually all software, well or poorly designed, far more approachable[18].

These advances being now available to us, we have entered a period in which the serious business of determining *how much money is needed* can now be taken care of with the same tools required to determine *what to do*.

[18] In 1981, when MIL-STD-1388-1A/B was published relational databases were largely a theoretical concept that would require another 10 years before they came into popular use. In part, they and other usability enhancements of software had to wait for the maturation of the personal computer user interface in the form of a relatively bullet-proof form of Windows, only reached at release 3.1. Even so, computer speeds and memory capacities did not achieve a practical level for significant computational problems until the Pentium series of chips were produced in the mid-1990s.

Robert A. Butler

Mr. Butler received his BS, MS and CPhil degrees in Economics from UCLA. He began his career at the RAND Corporation in 1965, where he worked in both the economics and logistics departments. In 1970 he began work at Technology Service Corporation, where he became Manager of the Systems Planning and Economics Division. He established Systems Exchange in 1976 as a defense consulting firm. Butler designed and wrote EDCAS and SDU, collaborated with Dr. Craig Sherbrooke on VMetric, designed and built the TFD Database and the first version of MAAP and has served as chief architect for the Tools for Decision software suite while leading the company to become a global enterprise of six companies with 120 people providing consulting and technical support services for over 5,000 software users at 600 licensed sites. Mr. Butler has published widely and has lecture associations with a number of Universities.

Contact Details:
Robert A. Butler,
Chairman, TFD Group
rab@tfdg.com
+1 831 649 3800
+1 831 649 3866 (fax)
P.O. Box 3290
Monterey CA 93942