2018 NDIA GROUND VEHICLE SYSTEMS ENGINEERING AND TECHNOLOGY SYMPOSIUM
MODELING & SIMULATION, TESTING AND VALIDATION (MSTV) TECHNICAL SESSION
AUGUST 7-9, 2018 - NOVI, MICHIGAN

GETTING AHEAD OF THE GAME: QUANTIFYING THE VALUE OF LABORATORY TESTING PRIOR TO FULL-UP SYSTEM LEVEL

Joel Luna
Sharon Snider
Frontier Technology, Inc.
Beavercreek, OH

Mark Brudnak, PhD
Bryan LaRose
Melissa Morgan
Dan Kosinski
Mike Scott
US Army TARDEC
Warren, MI

ABSTRACT

Program offices and the test community all desire to be more efficient with respect to testing but currently lack the analytical tools to help them fit early subsystem level testing into a framework which allows them to perform assessments at the system level. TARDEC initiated a Small Business Innovative Research (SBIR) effort to develop and deploy a system reliability testing and optimization tool that will quantify the value of subsystem level tests in an overall test program and incorporate the results into system level evaluations. The concept software, named the Army Lifecycle Test Optimization (ALTO) tool, provides not only the optimization capability desired, but also other key features to quickly see the current status, metrics, schedule, and reliability plots for the current test plan. As the user makes changes to the test plan, either by running the optimization or adjusting inputs or factors, the impacts on each of these areas is computed and displayed.

INTRODUCTION

The Program Managers (PMs), Program Executive Offices (PEOs), Research Development and Engineering Centers (RDECS), and Original Equipment Manufacturers (OEMs) and suppliers who develop and integrate technology into ground vehicles all perform tests on subsystems and components. While the Army depends on a system-level test to validate OEM compliance with reliability requirements, the PM/PEO and test community all desire to be more efficient with respect to testing. The automotive industry uses laboratory testing to great efficiency and effect, reducing cost, saving time, minimizing risk and elevating performance. OEMs, suppliers, RDECS, and other government and private labs are available and are used (at contractor discretion) to improve reliability and discover failure modes. The Army, however, currently lacks adequate analytical tools to help them fit early subsystem level testing into a framework which allows them to characterize improvements from subsystem test at the system level. The Army also lacks the tools needed to determine which subsystems should be tested in the lab in terms of a tangible return on investment of reduced cost, time and/or risk.

Research Effort Initiated

For this reason, the Tank and Automotive Research Development and Engineering Center (TARDEC) initiated a Small Business Innovative Research (SBIR) effort (Topic A16-085 titled “Lifecycle Test Optimization”) to develop and deploy a system reliability testing and optimization tool that will quantify the value of subsystem level tests in an overall test program and incorporate the results into system level evaluations. This effort included the development of a methodology with mathematical formulations and a software tool that implements the methodology. Results from the first phase showed that a rank ordered list of subsystems to test could be produced, along with corresponding overall test cost, time, and risk. A mathematical formulation for computing reliability growth at the subsystem level aggregated to the system level was also developed, as well as optimization methods for determining the optimal allocation of testing to produce the lowest cost and risk. These were implemented in a concept software tool, named the Army Lifecycle Test Optimization (ALTO) tool, which provides not only the optimization capability desired, but also other key features such as import of reliability design and test data and user friendly views to quickly see the current status, metrics, schedule, and reliability plots for the current test plan. The prototype version of ALTO is now in development to not only help with initial planning, but also updating with test results and re-planning, enabling the determination of the optimal use of remaining time and resources.

RELIABILITY CONSIDERATIONS

All major acquisition programs for the Army have reliability requirements that they must demonstrate during Operational Testing (OT) in order for the system to be deemed operationally suitable and in turn get fielded. The system is expected to exit Developmental Testing (DT) and enter OT with a high likelihood of demonstrating the system’s reliability requirement with high confidence. Performing reliability testing, especially early on to discover failure modes and allow time for adequate design fixes, increases the likelihood that the system will exit DT with a target Mean Miles Between Failures (MMBF) or better. This in turn increases the likelihood the system will demonstrate its reliability requirement to high confidence during OT. Since the Army’s method of determining confidence in OT is fixed (relying on a Chi-Square distribution assumption for uncertainty about the MMBF) [1], the only way to increase confidence in achieving the required reliability in OT as it is currently determined is to increase the expected MMBF coming out of DT. While there is value to the PM of having some idea of the uncertainty about the MMBF coming out of DT, this uncertainty characterization is not used to add confidence to the OT required MMBF given the current way that it is computed.

Reliability Growth

Because of the importance placed on demonstrable reliability, all major acquisition programs need to develop reliability growth plans early on and show progress towards following those plans throughout system development. These reliability growth plans are intended to spell out the various phases of system level DT that will be carried out and in turn show that with sufficient time allocated for fixing observed failures.

One of the major inputs to a reliability growth plan is the initial MMBF, which is the expected reliability that the system is predicted to have entering the first phase of DT. This assumed initial MMBF along with the assumed Management Strategy (MS) and Fix Effectiveness Factor (FEF) are used to compute the growth potential, which is the highest reliability that could be achieved if the system was tested an unlimited amount with fixes being incorporated throughout DT. That reliability ceiling goes up if the system enters DT with a higher initial MMBF but can also drop if the system enters DT with a lower initial MMBF.

Design for Reliability activities before DT has begun are important to reliability growth planning. These activities can include Failure Modes Effects and Criticality Analysis (FMECA) development, Physics of Failure analysis, subsystem level testing and other early reliability activities. These activities aim to identify failure modes early on at the subsystem and component level so they can be designed out of the system in a very cost effective manner. Performing these activities early on increases the likelihood that the system will enter DT with the assumed initial MMBF or better. This in turn increases both the expected reliability of the system at the end of DT and the reliability ceiling for the system which ultimately makes the system more likely to demonstrate its reliability requirement to high confidence during OT.

Failing to perform these activities, however, could increase the likelihood that the system starts DT at a reliability less than the assumed initial MMBF, which could lead to a higher risk of not demonstrating the system's reliability requirement during OT. The ALTO tool being developed helps to actually quantify the value of different subsystem level reliability testing options in terms of lowering the risk of not demonstrating the system's reliability requirements during OT by maximizing the MMBF achieved by the end of DT. As such, it could aid the reliability growth planning process by helping programs gain insight into the benefit of conducting the crucial Design for Reliability activities early on.

Reliability Growth Modeling

There are multiple reliability growth models that can be employed to plan and track reliability growth [2]. Most notably, the Army relies on the PM2 model for reliability growth estimation [3]. PM2 has significant limitations for application to ALTO, however. PM2 does not provide good guidance as to how much reliability growth testing is sufficient before the evaluation testing. For example, you can grow reliability a lot faster for a subsystem with relatively few failure modes than you can for a full system with many failure modes, but PM2 does not discern this. Also, PM2 does not lend easily to use in an optimization solution. For these reasons, PM2 was not adopted as the reliability growth model for ALTO, but PM2 growth curves will be displayed in ALTO to compare with the ALTO defined growth curves, since they use the same parameters.

Other Reliability Tests

ALTO also takes into consideration tests conducted for purposes other than reliability growth, such as durability tests. Even in cases where fixes are not implemented (as is the case in reliability growth), conducting such tests reduces uncertainty by obtaining actual failure rate performance for a number of test miles for specified test conditions. While reducing the uncertainty is not reflected in the confidence computation for the OT required MMBF, it does provide some confidence to the PM that the DT MMBF goal will be met.

ANALYTICAL FRAMEWORK

Based on the reliability considerations just discussed, the analytical framework that provides the computational foundation for ALTO can be introduced. It is comprised of the mathematical framework, consisting of equations that provide the links between inputs (parameters and factors) and outputs, as well as optimization algorithms that provide the recommended mix of testing to meet goals at the minimum cost (or other measure) subject to constraints.

Mathematical Framework

The ALTO mathematical framework first identifies parameters, factors, targets, and constraints, which were used as the basis for software concept development and case study execution. *Parameters* are the driving variables, or forcing functions, of the mathematical framework.

The first of these are the failure mode failure rates, which in the case study came from FMECA data, but could have come from any source of failure rate estimation at even a high level. The failure rates provide the basis for estimating initial reliability and reliability growth potential, as well as a number of other inputs. The other key parameter is test miles, which drives the reliability growth, time and cost, and risk at the OT level.

Factors are the variables that are applied to the parameters to affect outputs. For reliability growth, the key factors are MS and FEF. Time factors are set up time and recurring time (i.e., miles per active test day). Note that setup time results in a level for time and related cost 'buy-in' to overcome to conduct testing for each subsystem or the system. For cost, the key factor is cost per time unit, although this can easily be expanded to include other costs. For risk, the factors are Confidence and Probability of Acceptance (PoA) at the system OT level, which affect DT MMBF goals in order to achieve them. ALTO also includes adjustment factors from DT to OT to account for uncertainty going from DT conditions to OT conditions.

Targets, or goals, are the desired ends of the testing process, typically expressed in programmatic goals, such as OT level reliability, schedule, or cost. Other targets are also included, such as required MMBF at OT (OT level reliability goal for a given confidence, PoA, and test miles), achieved MMBF goal from DT, and lowest cost (or other objective) for achieved DT MMBF.

Constraints are limits on outputs computed from factors and parameters, such as budget (constraint on cost), schedule (constraint on test time), risk (constraint on OT level confidence and/or Probability of Acceptance).

Computation of Reliability Growth

The initial input failure rate parameters are used as the starting point for the reliability growth computations. The MS reliability growth factor is used to compute what fraction of failures can be surfaced that might be fixable (the B mode failure rate), and what proportion might actually be fixed with the FEF value. As testing (i.e., test miles) are projected, reliability growth can be computed at the component or subsystem level and then aggregated to the system level. The achieved reliability growth is determined as the amount of failure rate reduction for a given number of test miles, where the B mode failure rate that results from surfacing and fixing is dependent on the number of test miles. For each component or subsystem, the probability that a failure has been discovered is computed, and then used to compute an expected failure rate for that failure component or subsystem based on the FEF. An exponential failure time is assumed for the probability, which is consistent with PM2 as described in MIL-HDBK-189C [2]. A cumulative exponential distribution for each component or subsystem serves as the probability that a failure has been detected. Then the expected failure rate for the miles T is computed. This approach matches well with general expectations about how reliability growth occurs. For example, there is zero reliability growth for no test miles, and as test miles goes to infinity, the reliability growth approaches the reliability growth potential. This is shown in Figure 1, where failures are reduced as test miles are accumulated. With this equation,

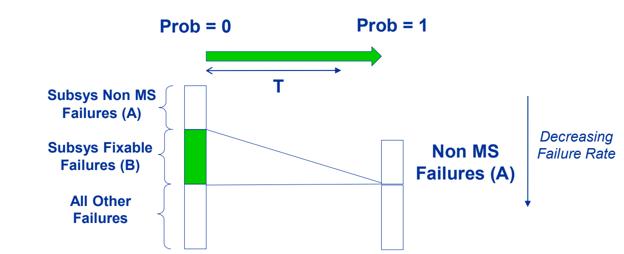

Figure 1. Increasing test miles reduces the failure rate, resulting in reliability growth.

one can obtain estimates of failure rate and MMBF that reflect reliability growth in failure modes as a function of test miles. Furthermore, estimates can be obtained for subsystem level reliability growth as a function of test miles and aggregated to determine the system failure rate. Test miles can be uniquely assigned to each subsystem for subsystem testing, while all subsystems have the same test miles for system level testing. Plots can then be obtained for system and subsystem MMBF versus test miles (T) as shown in Figure 2.

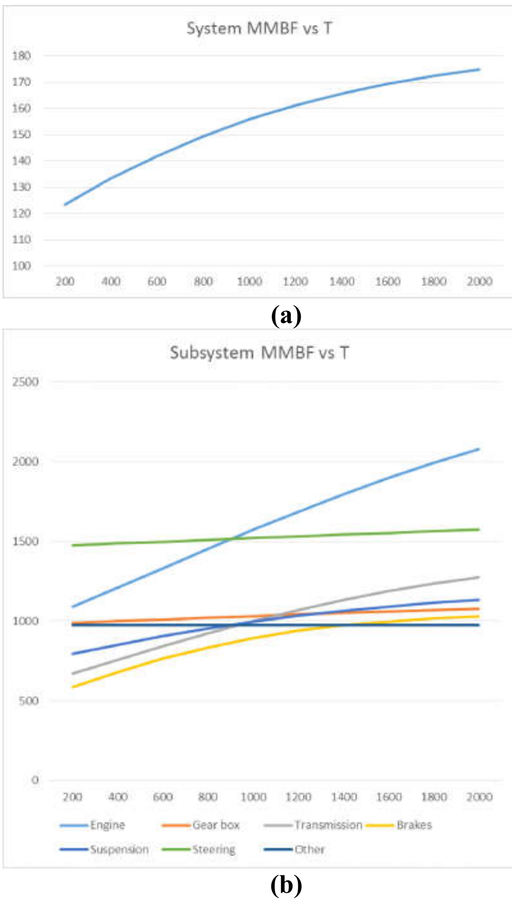

Figure 2. (a) System and (b) Subsystem MMBF versus Test Miles.

Test Miles as Dependent Variable and Relating to Program Information

By rearranging the expected value function for the failure rate and recognizing that the failure rate decrease is the same as the overall difference between the original and goal failure rates, an equation for miles as a function of failure rate change can be obtained. This enables plots to be obtained for system and subsystem miles as a function of reliability growth, most easily computed as differences between failure rates, as shown in Figure 3. The test miles obtained can then be used to compute program information such as schedule and cost for subsystems and the system.

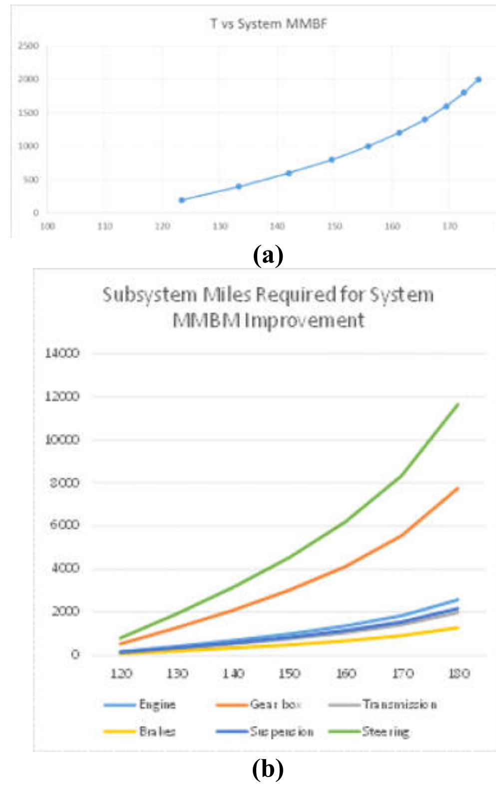

Figure 3. Reliability increments can be used to compute test miles versus MMBF for (a) system and (b) subsystems.

Schedule information is the calendar time to conduct testing at the subsystem and system DT level. The calendar time is divided into two parts – active test time, which is the time during which testing is being conducted, and corrective action time, which is the time when no testing is being conducted so that fixes can be implemented. *Active test time* consists of two parts – the one-time setup/teardown times that are required to conduct testing that are independent of test miles, and the recurring test time that is a function of test miles. The setup/teardown times are only incurred when testing is to be performed for a particular subsystem (or system), so that there is an initial time investment required to perform testing for any subsystem. The recurring test time is computed as the test miles per day divided by the test miles computed from reliability growth goals. The sum of these two is the active test time. It should be noted that subsystem test times are assumed to be conducted in parallel, so that the total calendar time is specified by the maximum subsystem test time.

Corrective action times are needed as inputs for each subsystem. The cumulative corrective action time after some test miles T is a function of whether the failure mode has been detected, so the computation depends on the cumulative probability it has been detected. For any given phase, the corrective action time that occurs in that phase is the expected cumulative corrective action time to the end of the phase minus the expected cumulative corrective action time to the end of the prior phase. The expected corrective action time for a subsystem in any phase is the maximum of the failure mode expected corrective action times for that phase. The expected overall (calendar time) expected corrective action time in any phase is the maximum of the subsystem expected corrective action times.

The total calendar time for testing and fixing is given by the sum of the active test time and the expected corrective action time. The total time is used to compare with the scheduled time to either enforce schedule constraints or to determine schedule impacts for planned testing strategies. FTI provides a calculator in the software tool to examine the effect on expected corrective action times and total test time by varying test miles in subsystem, system DT, and system OT phases.

Cost is computed by applying a cost factor to active test time, i.e., a cost per day multiplied by the number of days for active test, which includes both setup and recurring time. FTI's approach is based on the assumption that costs are most directly related to the time use of test facilities. Thus, to compute costs, first the test miles are determined, then the active test times, and then the costs. This approach can be easily revised to include one-time costs or base costs on a different basis other than active test days (such as test miles).

Optimization

In the course of this effort, FTI has developed three optimization algorithms for DT level. These algorithms are based on the equations described in the previous section to compute test miles T as a function of reliability increases, an optimization algorithm could be developed. Reliability increases are actually computed as failure rate decreases, since these are linear and even increments are more easily computed than with MMBFs. The active test time and cost values are then computed from the test mile values. Constraints to test miles, active test time, and cost values can then be applied to limit how the subsystems participate in the optimization.

The initial optimization algorithm determined optimal subsystem level testing based on minimizing cost, active test time, or test miles for even increments of reliability increase (failure rate decrease). This algorithm treated subsystems only. The test mile objective function values for each subsystem were built by first computing cumulative T values for even decrements of failure rate, and then computing the differences between cumulative T values to get the incremental increases corresponding to each failure rate decrement. It is important to do this since cumulative T values do not increase linearly for even decrements of failure rate. Active test time and cost objective functions were computed similarly by applying the appropriate factors to the cumulative T values and then computing active test time and cost increments. This way, initial setup times and costs are preserved in the optimization algorithm.

In each iteration of this algorithm, the incremental increases in the optimization basis (i.e., test miles, active test time, or cost) are then compared between subsystems to determine which has the least increase for the same failure rate decrease as shown in Figure 4. This results in the lowest possible values for each iteration, so that optimality is ensured over all iterations. When the least increase is identified, that increase is added to the total for that subsystem, and the next increment is used in subsequent iterations. When a constraint in test miles, active test time, or cost for any subsystem is exceeded, that subsystem is no longer considered in subsequent iterations. The algorithm continues until the total failure rate decrements have been accomplished (corresponding to the goal MMBF improvement) or all subsystems are constrained. The result of the algorithm is the total test miles, active test time, or cost by subsystem for a range of system MMBF values, which are computed from the failure rate decrements.

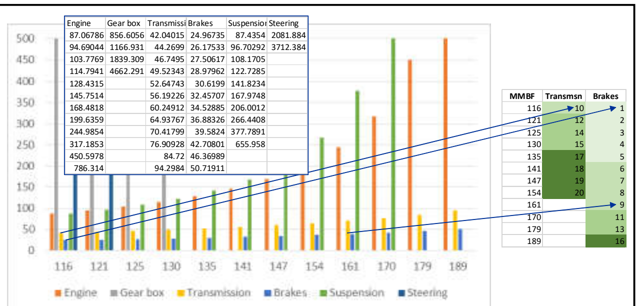

Figure 4. Optimization is performed by selecting the lowest cost test option for reliability improvement increments.

Because the original optimization algorithm did not include system level DT with the subsystems so that a balance between system and subsystem level testing could also be obtained, FTI enhanced the algorithm to include system level testing, which required two significant changes. First, the objective function values could not be computed beforehand because the subsystem test mile, active test time, and cost values are dependent on system level values, while previously the subsystem values were independent of each other. Therefore, FTI revised the algorithm to update the cumulative values prior to evaluating the next failure rate decrement. Second, the value of system test miles to achieve a failure rate decrement cannot be analytically computed because it requires knowledge of how the failure rate decrement would be allocated among the subsystems to achieve the total system failure rate decrement (assuming that each subsystem is exposed to the same number of test miles since they are all part of the system when it is being tested). Therefore, a search algorithm is required to be used each iteration to determine what system level test miles will produce the desired MMBF improvement. Time and cost factors are also required now for the system level DT to compute active test time and cost increments. With these enhancements, the algorithm searches for the lowest increase for the same failure rate decrement, where the system is considered in each iteration along with the subsystems. The interface for the optimization algorithm in ALTO is shown in Figure 5.

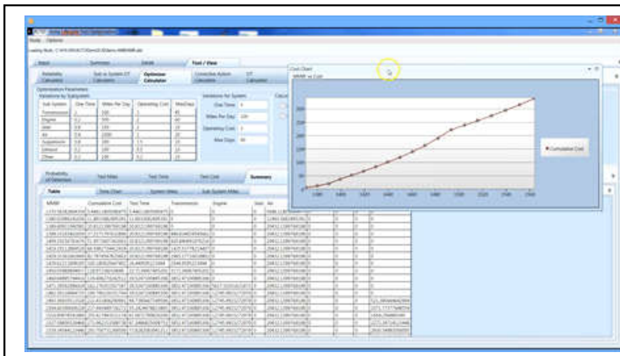

Figure 5. Results of the Optimizer show the progression of optimal allocations among subsystem and system level development testing.

The algorithm in the base effort assumed a single optimization objective, i.e., either test miles, active test time, or cost. This was extended subsequently to multi-objective optimization, where the same algorithm is followed, but the objective function values are combined with weights to reflect multiple objectives. This algorithm was implemented in the concept software in the option effort.

Simulation

Simulation was used to validate analytical estimates based on the same assumptions, which metrics were the most helpful, and how uncertainty could be characterized [4]. The simulation model separated the generation of A mode failures from B mode failures by creating separate failure generating processes using an exponential distribution between failures based on separate failure rates (corresponding to the value of MS). A mode failures were repaired with no change to the failure rate. B mode failures when first occurring were evaluated for fix effectiveness. A random draw was made to determine if a modification would eliminate the failure mode, and if so, the failure did not occur again (emulating a decrease in the failure rate since each subsequent failure did not result in maintenance, and thus effectively was not counted as a failure). If the failure mode was not eliminated, the failure mode would subsequently be treated as an A mode failure. Delaying modifications (and thus failure rate reductions) to corrective action periods was also incorporated but not exercised as part of the evaluation of the simulation.

The probability of detection, number of failures, and Mean Miles Between Failures (MMBF) were computed with simulation results. The probability of detection was computed as the sum of detections (a detection was considered as the first failure that occurred and was output as a binary value – one for a detection occurred and zero if not) divided by the number of iterations across all failure modes for the system. The number of failures was output as a cumulative count for each day (at a constant number of miles per day) and for the duration of the simulation for each iteration across all failure modes for the system. The mean number of failures was computed as the total divided by the number of iterations. Cumulative and (relatively) instantaneous MMBF values could then be computed based on the mean number of test miles (total test miles across the number of iterations divided by the number of iterations) and mean number of failures and used to compare to the analytical estimates.

Two simulation results were compared with analytical estimates. The first is an estimate of the instantaneous MMBF using simulation probability of detection values and the second is a computation of cumulative MMBF. These are shown with the analytical estimate in Figure 6. As can be seen, there is very good agreement between the analytical estimate and the estimate computed from simulation probability of detection values ('M(T) Est from Subsys'). The cumulative MMBF estimate from simulation is lower, which is not surprising since it is a cumulative versus an instantaneous value, but further investigation also revealed that capping input failure rates may overall be shortening time between failures, resulting in a higher number of failures from simulation, resulting in a lower simulation MMBF estimate.

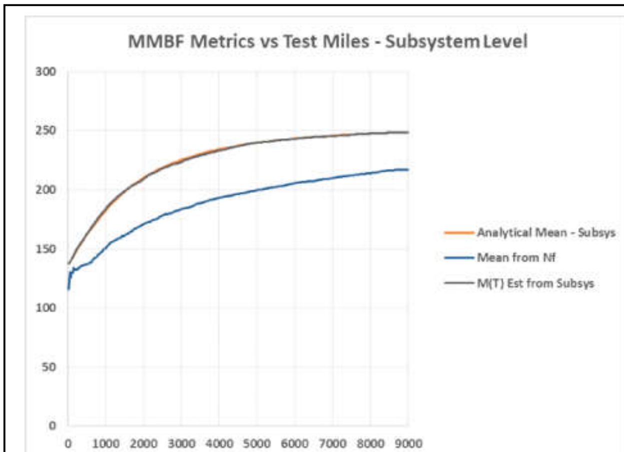

Figure 6. Comparison of MMBF estimates from analytical and simulation computations.

ALTO

The ALTO application is a Windows based desktop application developed by Frontier Technology Inc. (FTI) to quantitatively define the value of subsystem level tests and incorporate the results into system level evaluations. ALTO provides the ability to relate component and subsystem tests to system level reliability and optimize the amount of subsystem level testing and can be used to develop test plans that most effectively test the right variants of vehicles in the most productive terrains. The top level dashboard view is shown in Figure 7.

Figure 7. The ALTO Dashboard view provides a quick status review of test plan metrics, schedule, and reliability growth.

Components

The components of the ALTO prototype software are enumerated below.

Study. Accessed by clicking the Study menu tab. It contains functions for study opening, saving, and an about ALTO view.

Input. Accessed by clicking the Input menu tab. It contains functions for importing test data files and setting test factors and constraints used in computing test measures.

Configure. Accessed by clicking the Configure menu tab. It contains functions for configuring default test factors and constraints for user's typical test evaluation requirements.

Reliability Calculator. Accessed by clicking the Reliability Calculator menu tab. It contains functions for adjusting reliability factors used in computing reliability test measures.

Sub vs System DT Calculator. Accessed by clicking the Sub vs System DT Calculator menu tab. It contains functions for comparing and adjusting subsystem to system development test reliability factors used in computing reliability test measures.

Corrective Action Calculator. Accessed by clicking the Corrective Action Calculator menu tab. It contains functions for adjusting corrective action or fix factors used in computing time and cost test measures.

Optimizer Calculator. Accessed by clicking the Optimizer Calculator menu tab. It contains functions for managing selection of optimal test plan factors.

OT Calculator and Utility. Accessed by clicking the OT Calculator or Utility menu tab. It contains functions for managing factors used in computing operational test reliability, cost, time, miles, and risk measures. It also contains functions for viewing the effects of changing different operational test factors.

Message Pane. This pane, located at the bottom on the window, displays information and error messages pertinent to the user as the ALTO application is executing.

Dashboard Tabs. These tabs (and sub-tabs) provide access to view and manage test scenario measures.

Dashboard. Provides a quick view of measures resulting from the application of the current data inputs, factors and constraints.

Details. Provides a more detailed summary and listing of the current test plan measures.

Simulation (TBD). Allows users to refine data sets via the use of simulation and view simulation results, as well as manage updates to the test plan based on the simulation execution.

CONCEPT OF USE

A general methodology was developed to provide a process that leverages the analytical framework described in the previous section. In order to be effective, a concept of use for these capabilities needs to be developed to enable the support of test planning. The general methodology is shown in Figure 8 and described further below.

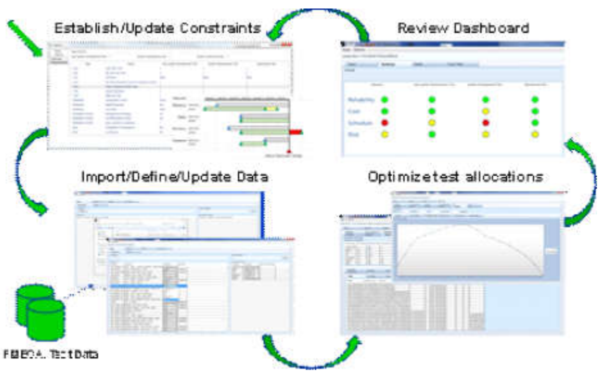

Figure 8. The process of using ALTO is expected to establish initial test plan parameters and then update continuously throughout the test program.

Initial Planning

Import/enter failure data. This data is comprised of the failure rate data that can be obtained, whether a high level or a very detailed level (e.g., FMECA data). A data import function was implemented in the concept software to facilitate this step. It may be that the failure rate data is not related to a component or subsystem, so an assignment function in the concept software was implemented to facilitate this step.

Enter reliability growth factors. In this step, the user needs to define MS and FEF values to model reliability growth, so the software enables the definition of MS and FEF defaults at the system level.

(Optional) View measures and explore changes. The test planner may well want to view subsystem and system level measures based on the imported data and any edits the user has made, as well as explore how measures change due to changes in test miles or changes in reliability growth factors. For this reason, the capability is provided to view initial measures and MMBF versus T in the concept software to see how changing system and/or subsystem miles changes MMBFs and other measures.

(Optional) Set initial conditions for DT subsystem or system. The user may well want to set initial conditions regarding test miles based on experience, expert opinion, or update the plan with actual results. The user is enabled to set initial miles for test planning before performing optimizations and DT vs OT tradeoffs.

Enter programmatic inputs. In this step, the user now adds additional inputs for primarily programmatic information, such as time and cost factors and constraints on subsystem or system level DT. As with MS and FEF factors, the user is enabled to specify default time and cost factors that can be tailored by subsystem.

(Optional) View programmatic measures. Once the programmatic inputs are provided the user can view costs, active test time, corrective test time, and total test time vs test miles. The user also has the option of viewing measures based on these factors, and explore the impacts on overall measures of changing the values of those factors.

Determine optimal DT balance between subsystem and system level testing. The initial approach enables the user to optimize on cost, active test time, or test miles. This approach has been expanded to include multi-objective optimization, such as maximizing reliability and minimizing corrective action time. The ALTO software will enable the user to inspect how constraints affect the system or subsystem optimization values.

Examine corrective action time reduction as needed. The user can also evaluate the expected corrective action times in each phase, and to see which of the subsystems are driving those times. The user can then make changes to subsystem miles to reduce subsystem corrective action times as needed, and to view the impacts on corrective action, active test, and total time, as well as cost.

Updating with Test Results

As test results become available, a means of computing MMBF estimates from test data with uncertainty, and comparing them with initial MMBF estimates (e.g., from FMECA estimates), will provide insight into how well the reliability plan is being implemented (i.e., whether the reliability goal is on track to be achieved) and how well the reliability is characterized (i.e., reducing uncertainty of the estimate). This area is currently being researched, including innovative methods [5], and developed for implementation in the ALTO tool. The ability to read and process Test Incident Report (TIR) data from the Vision Digital Library System has been demonstrated, with the next step being incorporation of that data into an overall comparison framework between test and design reliability estimating data.

Re-planning

As reliability estimates are updated, the initial conditions can be reset to reflect progress made and subsequent test optimized as needed to ensure the reliability will be achieved as planned. While the ALTO tool facilitates subsequent re-planning by enabling the user to easily modify initial conditions, the overall approach to re-planning is still be refined.

CONCLUSIONS AND NEXT STEPS

While significant progress has been made to develop a software tool to aid in showing the value of laboratory testing prior to full-up system level test, there are a number of areas that require further development. First, a consistent framework for merging test and design reliability estimate data in terms of designating failures as well as computing and comparing uncertainty is being developed and refined. Second, the ALTO tool is being refined to facilitate quick inspection of progress on achieving reliability, and investigation into causes if it appears that the reliability objective will not be achieved with sufficient certainty. Third, the characterization of uncertainty reduction through test is also being refined, and incorporated into the multi-objective optimization, so that even in cases where reliability growth is not achieved, the reduction in uncertainty can be a basis for making certain subsystem tests more desirable than others. As this effort progresses, the ALTO software, underlying analytical framework, and case study execution will be matured and increasingly introduced into Army organizations as well as OEMs to facilitate the most cost-effective approaches to subsystem and system test planning possible.

REFERENCES

[1]D. Kosinski, “Advantages of Simulation Based Reliability Growth Planning”, SAE Technical Paper 2015-01-0439, 2015.

[2]MIL-HDBK-189, “Military Handbook: Reliability Growth Management”, 1981.

[3]D. Kosinski, “Simulation Based Reliability Growth Planning Model for the Army”, Proceedings of the 2014 Ground Vehicle Systems Engineering and Technology Symposium (GVSETS), Novi, MI, 2014.

[4]J. Luna, Final Technical Report, Lifecycle Test Optimization, SBIR Topic A16-085, 2018.

[5]K.K. Choi, H. Cho, M. Moon, N. Gaul, D. Lamb, and D. Gorsich, “Confidence-Based Uncertainty Quantification and Reliability Assessment Using Limited Numbers of Input and Output Test Data and Validated Simulation Model”, Proceedings of the 2017 Ground Vehicle Systems Engineering and Technology Symposium (GVSETS), Novi, MI, 2017.