A Simulation and Modeling Based Reliability Requirements Assessment Methodology

Conference Paper · August 2014

DOI: 10.1115/DETC2014-35482

8 authors, including:

Tomonori Honda
Palo Alto Research Center
41 PUBLICATIONS **122** CITATIONS
SEE PROFILE

Eric Saund
Palo Alto Research Center
69 PUBLICATIONS **1,260** CITATIONS
SEE PROFILE

Ion Matei
Palo Alto Research Center
42 PUBLICATIONS **250** CITATIONS
SEE PROFILE

Daniel G. Bobrow
Palo Alto Research Center
234 PUBLICATIONS **13,006** CITATIONS
SEE PROFILE

Proceedings of the ASME 2014 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference
IDETC/CIE 2014
August 17-20, 2014, Buffalo, USA

DETC2014/DTM-35482

A SIMULATION AND MODELING BASED RELIABILITY REQUIREMENTS ASSESSMENT METHODOLOGY

Tomonori Honda, Eric Saund, Ion Matei, Bill Janssen,
Bhaskar Saha, Daniel G. Bobrow, Johan de Kleer, Tolga Kurtoglu
Intelligent Systems Laboratory
Palo Alto Research Center, Inc.
Palo Alto, CA
Contact Email: Tomo.Honda@parc.com

ABSTRACT

To minimize the design cost of a complex system and maximize performance, a design team ideally must be able to quantify reliability and mitigate risk at the earliest phases of the design process, where 80% of the cost is committed. This paper demonstrates the capabilities of a new *System Reliability Exploration Tool* based on the improved simulation capabilities of a system called *Fault-Augmented Modelica Extension* (FAME). This novel tool combines concepts from FMEA, traditional Reliability Analysis, and Quality Engineering to identify, gain insight, and quantify the impact of component failure modes through time evolution of a system's lifecycle. We illustrate how to use the FAME System Reliability Exploration Tool through a vehicle design case study.

NOMENCLATURE

d damage amount
$d^c_{i,j}$ critical damage amount for component failure mode i and performance metric j
d^c_i critical damage amount for component failure mode i
$p_{i,m}(d)$ probability density function for amount of damage d after m missions
$p^f_{i,m}$ Probability of Mission Failure after m missions caused by component failure mode i.
p^f_m Probability of Mission Failure after m missions for a particular system configuration

MOTIVATION

The DARPA Adaptive Vehicle Make (AVM) program aims to compress the military vehicle design, development, test, and evaluation cycle by a factor of 5 [1]. When a system is deployed, faults occur in the field due to harsh environmental and operational conditions, poor maintenance, and attacks on equipment. Therefore, it is crucial to reason about fault behavior and failure impact as part of early system design exploration and analysis. Just as reliability is key for the systems that support the modern warfighter, it is equally important for systems used in civilian life. Examples include transportation systems, medical facilities, and HVAC equipment. Modern cyber-physical systems feature a high level of complexity that renders the traditional design methodology of build-test-evaluate-redesign highly inefficient. Often the critical nature of these cyber-physical systems mandates a thorough analysis to fully understand and quantify component faults and their impact on system behavior. A key technical challenge in developing such complex systems is to ensure that individual components are reliable under realistic usage and the overall system is functionally robust under component failure, resulting in reliable designs. This requires the integration of reliability analysis into the system design process as early as possible.

There has been much discussion about conflict between design theory developed by design researchers, and design tools used by design practioners [2, 3, 4]. Ideally, design tools should be based on rigorous design theory, while remaining straightfor-

ward enough to be used by practioners. The new Fault Augmented Modelica Extension (FAME) based System Reliability Exploration Tool estimates reliability in term of theoretically-supported system performance metrics, while providing intuitive insight to designers, enabling them to choose wisely among system configurations, understand critical component failure modes, and outline maintenance scheduling.

BACKGROUND

The FAME based System Reliability Exploration Tool captures reliability in early stage design by combining Quality Engineering, Reliability Engineering, and Design Theory. The new tool aggregates stochastic damage acculumation for each failure mode with cyber-physical simulation to generate and display a time evolution of engineering performance. Unlike prior work [5,6,7], this reliability exploration tool focuses on impact of single failure modes separately, while providing an intuitive user interface to explore design trade-offs. Finally, this FAME based System Reliability Exploration Tool is developed to answer the following type of questions:

- What *System Configurations* are most reliable?
- Which *Components* are most suspetible to wear/damage that causes critical performance loss?
- What *Performance Metrics* are most at risk?
- How do these factors vary with *operational time*?

Other quality/robustness/reliability approaches answer some of these questions, but not all of them. The new reliability assessment tool emcompasses both traditional robustness and reliability analysis, while utilizing the same model fidelity and simulations used to evaluate a nominal design (see Figure 1).

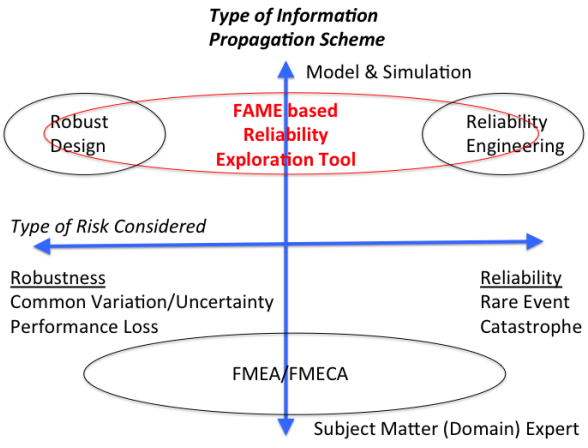

FIGURE 1. Comparison of FAME based Reliability Exploration Tool with other Traditional Techniques

A common approach taken by a design team to capture reliability in early design stages is Failure Mode and Effects Analysis (FMEA) and Failure Mode, Effects, and Criticality Analysis (FMECA) [8, 9], which had been the U.S. Department of Defense's standard [10] for reliability analysis until 1998 but remains commonly used by military and space applications. FMEA and FMECA quantify the likelihood and impact of each failure mode and guide the design effort to mitigate risk associated with critical failure modes.

Reliability Engineering is a particular sub-field of Systems Engineering that deals with requirements specifications for, and failure analysis of, systems. "Reliability" describes the ability of a system or component to function under stated conditions for a specified period of time. It encompasses costs of failure caused by system downtime, cost of spares, repair equipment, personnel, and cost of warranty claims. Effective reliability engineering requires basic understanding of failure mechanisms, which derives from knowledge and experience with different specialized fields of engineering, including: Tribology-, Stress-, Fatigue-, Thermal-, Electrical- and Chemical Engineering. Traditional requirement analysis is mainly based on Fault-Trees [11, 12, 13] which are graph-based models for representing various combinations of faults that will result in the occurrence of predefined undesired events. Another approach is based on Reliability Block Diagrams [14] which define the logical interactions of failures within a system through the depiction of network relationships between blocks. There are many other variants such as Dynamic Fault Trees [15] and Petri Nets [16, 17, 18]. These techniques have several important limitations in dealing with complex engineering systems. Such limitations include adopting a set of simplifications and assumptions that increase the distance between the model and the system under study (e.g., all failures are independent, the failure rates are constant), and having a causal representation of the system detached from physical meaning since most physical systems such as electrical, mechanical or thermodynamical systems are acausal in nature. Model-Based Reliability Analysis (MBRA) is a reliability analysis technique that capitalizes on insights gained from modeling to make both qualitative and quantitative statements about product reliability [19]. It has at its core System Modeling, which is the interdisciplinary study of the use of models to conceptualize and construct representations of systems; this discipline has appeared as a result of the development of System Theory and Systems Sciences. System Modeling is based on a modeling language, which is an artificial language used to express information about a system in a structure that is defined by a consistent set of rules. An increasingly popular language for describing systems is the Modelica language [20]. The Modelica language is used to express mathematical models of system behaviors from a wide range of engineering domains.

Quality Engineering, which is the basis of Robust Design, was made popular by Genichi Taguchi and focuses on improving

"quality" of the design [21, 22, 23, 24]. These robust design approaches have been developed to make engineering performance of a nominal design be robust to probabilistic uncertainties such as manufacturing variation and operational variations. Recently, the Robust Design approach has been extended to Probabilistic Certificate of Correctness (PCC) [25, 26, 27] to prune poor designs early in the design process.

A few other reliability/risk based design approaches developed by the design theory community are especially notable. Function-Failure Design Method [28] (FFDM) aids designers by predicting likely failure modes from intended product functionality. Risk in Early Design (RED), which capture risk in a Fever Chart, can be used to help visualize a set of system level risk events [29, 30, 31, 32, 33]. Graph-Based Fault Identification and Propagation [34] and Functional Failure Reasoning [35] extends FFDM to capture fault propagations. Reliability Based Design Optimization (RBDO) [36, 37, 38] captures the robustness of a nominal design and treats reliability as probabilistic constraints. Fleet Maintenance Simulation [39] has been funded by the US Army Tank Automotive Research, Development and Engineering Center (TARDEC) to enhance failure and maintenance analysis by (a) utilizing information theory to predict failure probability distributions, (b) propagate component binary failure information into system reliablity using fault tree and Monte Carlo Simulation, and (c) determine the Pareto Frontier for trade-offs between reliability and cost. These approaches tackle reliability at various design stages and from different perspectives.

FAME APPROACH

The FAME System Reliability Exploration Tool operates in three distinct steps.

1. Determine a stochastic process, called the *Damage Parameter Map*, associated with different failure modes.
2. Simulate the impact of damage and failures using the Fault Augmented Modelica Extension (FAME).
3. Aggregate the Damage Parameter Map and Simulation results, and display results and insights to the designer in a coherent manner.

Damage Parameter Map

Each of a system's components degrades with use over time through different physical mechanisms including fatigue, stress rupture, frictional wear, etc. [40, 41] Due to modeling uncertainties, manufacturing variability, and environmental uncertainties such as temperature, loading conditions, etc., the amount of a component's degradation can only be quantified in probabilistic space and should be modeled as a random process. Damage accumulation is estimated through stochastic process simulations using context models that incorporate load events, load histories, ambient chemical environment, temperature environment, and additional parameters to be specified. The simulations execute service duty scenarios that impose loads that cause stress and damage. The loads are functions of terrain profiles, impacts, or kinetic blast.

Generic physics-of-damage models are used to generate *damage parameter maps* for all component classes. A damage parameter map expresses a component's functional performance quantitatively as a function of a time parameter which could be based on clock time or usage. Damage parameter maps for different components can share underlying physical models. For example, the same basic fatigue damage model is used for gear tooth fatigue, driveshaft fatigue, or fatigue damage of other component classes. This is made possible by component class parameter transformations that use such attributes as dimensions and geometry, in conjunction with reference loads, to calculate such parameters as stress or strain that are used in the generic damage process models. Materials properties relevant to a specific failure mechanism are required for the material from which a component is fabricated.

Engine Bearing Example

Bearing failure in rotating machines has been studied for decades [42]. Each of the different causes of bearing failure (wear, surface distress, corrosion, electric discharge, etc.) produces its own characteristic damage [43]. Such damage, known as primary damage, gives rise to secondary, failure-inducing damage like flaking and cracks. This example deals with change in bearing geometry due to friction wear in an engine application. The bearing considered is a journal bearing fitted between the crankshaft and the main bearing housing. The wear depth h_{wear} incurred after running the bearing over time t is given by:

$$h_{wear} = kPVt \tag{1}$$

where, k is the specific wear rate, P is bearing pressure, and V is sliding velocity.

The corresponding radial clearance between journal and bearing, C, is given by:

$$C = C_0 + h_{wear} \tag{2}$$

where, C_0 is the initial clearance.

The Sommerfeld number, S, which is a non-dimensional design parameter involving the geometrical and operating features of the bearing, is given by [44]:

$$S = \frac{\eta NLD}{F}\left(\frac{D/2}{C}\right)^2 \tag{3}$$

where, η is lubricant viscosity, N is crankshaft speed, L is bearing length, D is bearing diameter, and F is bearing load.

The minimum film thickness, h_{min}, and the film thickness parameter, λ, are then given by [44]:

$$h_{min} = 4.678C(L/D)^{1.044}S \tag{4}$$

$$\lambda = \frac{h_{min}}{\sqrt{R_j^2 + R_s^2}} \tag{5}$$

where, R_j and R_s represent the journal and shaft surface finishes (RMS) respectively.

The worst-case coefficient of friction, μ, is approximated from the Stribeck curve [45] as:

$$\mu = a_0 \left(1 - \frac{1}{1 + e^{-a_1(\lambda - a_2)}}\right) + a_3 \tag{6}$$

where, $a_{0\ldots3}$ are regression coefficients.

The damage-parameter map for μ is derived by assuming tolerance distributions over the material and geometric properties of the bearing and sweeping the running time over 1000 hours.

Simulation using Fault Augmented Modelica Extension

The numerical simulations executed for reliability analysis use mathematical models of the system expressed through the Modelica language. We augment Modelica models with the capability to simulate failure mechanisms of two kinds [46].

Continuous damage accumulation produces gradual loss of performance over time or service cycles due to fatigue, wear, and corrosion. To model continuous damage accumulation, we augment Modelica component models to include *parametric faults*, as described in the next section. Gradual loss of performance in components may also be caused by damage accumulation at connection points. For example, in mechanical components, loss of lubricant in bearings generate loss of performance due the presence of an additional frictional phenomenon. Since Modelica models do not model behavior at the level of connectors, we address this by adding equations to Modelica components.

Qualitative failure damage reflects complete functional loss of a component. This could arise from a singular event such as impact, shock, or blast that pushes stress beyond strength limits, or it could result from an accumulation of damage that impacts component performance catastrophically, such as a fatigue failure. Qualitative failure mechanisms are modeled through changes in power transfer at connection points; interruption is implemented through the introduction of equations that simulate this phenomenon. We call these *power faults*. These equations serve the additional purpose of allowing us to abstractly model loss of performance in the absence of information about the failure mechanisms.

Our approach to fault augmentation relies on understanding the patterns of the power flows through the connectors of the components of a system and the parameters that determine the behavior of the components. Once the pattern is determined, a set of equations involving the variables of the connectors is automatically added; these equations are designed to be compatible with the physical domain the components operate in and with the type of fault modes we want to include.

Power fault injection relies on identification of standard power interfaces in the model. These are instances of Modelica connector classes, such as `Modelica.Electrical.Analog.Interfaces.Pin`, which contain two variables, one of an "effort" type, such as `Voltage` or `Pressure`, and another of a "flow" type, such as `Current` or `MassFlowRate`. The analysis examines both direct components of the model class, and components of each inherited class used by the model class. Parametric faults are handled by defining a new model for the corresponding parameter, which provides a transformation of that parameter by the specified function, modified by the amount of the fault, and then introducing a component of that type. The original parameter is connected to the input of this transformer component, while equations which referenced the original parameter are changed to reference the transformed output of the component. A comprehensive description of our augmentation methodology can be found in [46].

The fault augmented Modelica models can replace their nominal counterparts and provide the capability to test the effects of components faults in the overall behavior of the system. When aggregating large numbers of fault augmented models, the number of possible fault operating modes can be very large. To help with scalability, when testing the effect of single faults, we developed an automated mechanism that decreases the number of fault modes that need to be considered. We eliminate faults that appear in subcomponents that, due to a particular configuration of the system, are never activated; fault modes that, due to particular type of interconnection, cannot physically appear; and fault modes whose effects on the behavior of the system are identical.

Aggregation to System Reliability

In general, "reliability" describes the ability of a system to operate while meeting all requirements for a specified period of time. Reliability is often quantified in terms of likelihood of failure, e.g. Mean Time to Failure (MTTF), Mean Time between Failure (MTBF), and Failures in Time (FIT) which actually cap-

tures system *un*reliability. The FAME System Reliability Exploration Tool captures reliability using *Overall Probability of Mission Failure*, which is equivalent to "System Failure Probability" [47].[1]

This Overall Probability of Mission Failure aggregates information from Damage Parameter Maps and simulation results based on FAME models in the following steps:

1. Determine critical damage value for each component failure mode.
2. Compute component failure probability for the given number of missions (or operational time).
3. Aggregate component failure probabilities into overall probability of mission failure.

Determining critical damage for each component failure mode

Performance metrics are individual measures of system level performance pertaining to mission success. For example, performance metrics for a vehicle include maximum speed, time to accelerate to a certain speed, and stopping distance from a certain speed.

Critical damage is defined as the minimal damage amount that results in failing any required performance metric. The critical damage for any component failure mode and performance metric is determined by interpolating Modelica simulation results. Graphically, critical damage for a particular performance metric is represented by the damage amount at which simulated performance intersects the required performance as a function of damage amount (see Figure 2).

Note that if the simulated and required performance do not intersect, then (a) either critical damage is 0 which means that the nominal design doesn't meet the performance metrics, or (b) the selected component's failure modes are not critical to that paricular performance metric.

Critical damage for a particular performance metric is denoted by $d^c_{i,j}$, where i represents component failure mode and j represents performance metric. The critical damage amount for a component is determined by the first performance metric that fails as a result of this damage:

$$d^c_i = \min_j d^c_{i,j} \tag{7}$$

Computing component failure probability for the given number of missions

By fixing the number of missions m for the damage parameter map (see Figure 3), we can obtain a probability density function for the amount of damage d after m missions, $(p_{i,m}(d))$.

[1]Because the FAME System Reliability Exploration Tool has been developed for military vehicle design, this paper uses the following interchangable terminologies: a) *Operational Time* is captured by *Number of Missions*, and b) *system failure* is equivalent to *mission failure*.

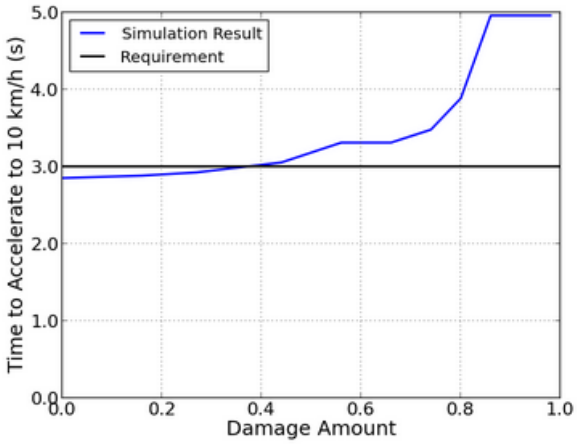

FIGURE 2. PERFORMANCE VS DAMAGE AMOUNT

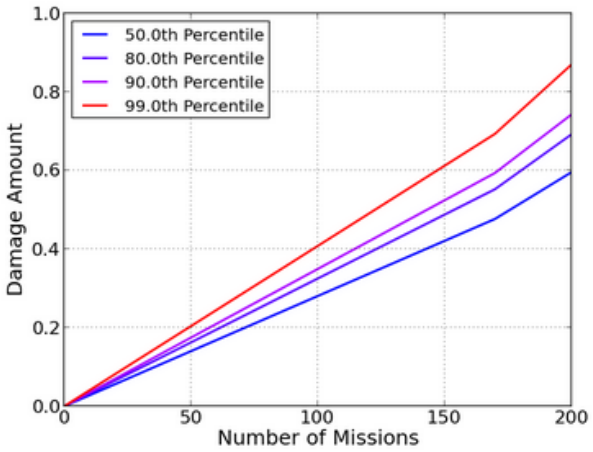

FIGURE 3. DAMAGE DISTRIBUTTION AS FUNCTION OF TIME (NUMBER OF MISSIONS)

Component failure probability is the probability that damage exceeds the critical amount:

$$p^f_{i,m} = p_{i,m}(d \leq d^c_i) \tag{8}$$

Note two special cases: (a) If d^c_i is undefined because component failure mode i is non-critical for all performance metrics, then there is no critical damage amount and the probability of component failure is always 0. (b) If d^c_i is 0 for component failure mode i because the nominal design fails even under no damage, then probability of component failure is always 1.

For a selected component subject to failure, the probability of meeting the selected required performance metric is obtained by combining the critical damage amount with the probability of incurring that much damage after the given number of missions.

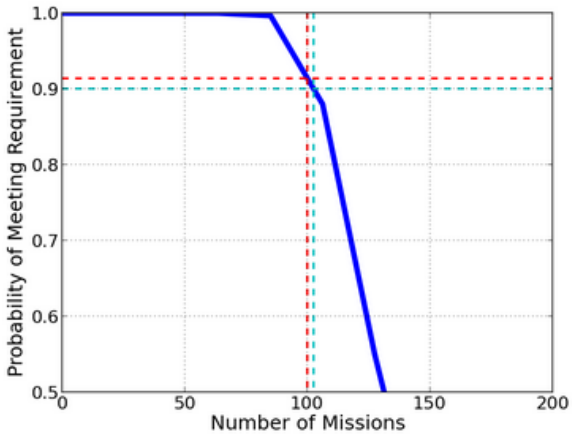

FIGURE 4. PROBABILITY OF MISSION SUCCESS AS FUNCTION OF TIME (NUMBER OF MISSIONS)

Graphically, this is shown in the Figure 4. The locus of where the target number of missions (vertical dashed red line) crosses the probability of meeting performance metric required value (dark blue line), is where one can read off the probability of meeting the performance requirement (horizontal dashed red line). Note that the tool can also be used to compute the maximum number of missions that satisfies the desired probability of mission success. This is indicated by the cyan line.

Given the per-component list of probability of meeting a given performance metric, a designer can determine critical component failure modes (see Figure 5). The critical component failure modes are functions of the number of missions, and with the appropriate UI tools, a designer can explore how each component failure mode impacts reliability at different numbers of missions.

Aggregating component failure probabilities into overall probability of mission failure.

By assuming that component failure modes are independent of one another[2] we can aggregate the component failure probabilities into an overall probability of mission failure by noting that mission failure is the inverse of mission success, and mission success is the conjunction of the conditions that all components

[2]This independence could be relaxed if the designer knows the degree of coupling among component failure modes.

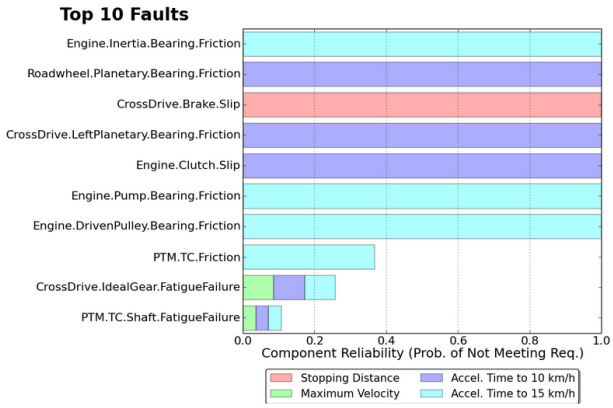

FIGURE 5. PROBABILITY OF MISSION FAILURE BREAKDOWN

not cause failure for any performance metric:

$$p_m^f = 1 - \prod_i \left(1 - p_{i,m}^f\right) \tag{9}$$

Given this formulation, usually only a few component failure modes dominate the possible causes for system failure.

Implementation and User Interface

The System Reliability Exploration Tool has been implemented through a web interface linked to a back-end service. The back-end service runs Python scripts to calculate performance metrics and generate results and graphs. The interface, shown in Figure 7, exposes all user-selectable parameters, along with computed values and graphs, in a dashboard format. The UI includes four major selectable parameters, and four graphs. The major selectable parameters are:

- *System Configuration:* A given design may allow variants on parts, modules, or components. In the "no_controls" example, system configurations reflect different combinations of vehicle engines, drive trains, and transmissions.
- *Component Fault:* A selector widget allows selection of a system component that is subject to fault, where the fault could be continuous damage due to age or wear, or else qualitative failure. In general, any given system level component will be subject to multiple faults related to its internal workings. In the current implementation, at most one component fault can be selected at a time.
- *Number of Missions:* The designer can effectively explore the time course of fault consequences by adjusting effective system age, which for vehicles is termed, "Number of Missions."
- *Performance Metric:* Each performance metric includes a *Performance Requirement* (e.g. time to accelerate to 15

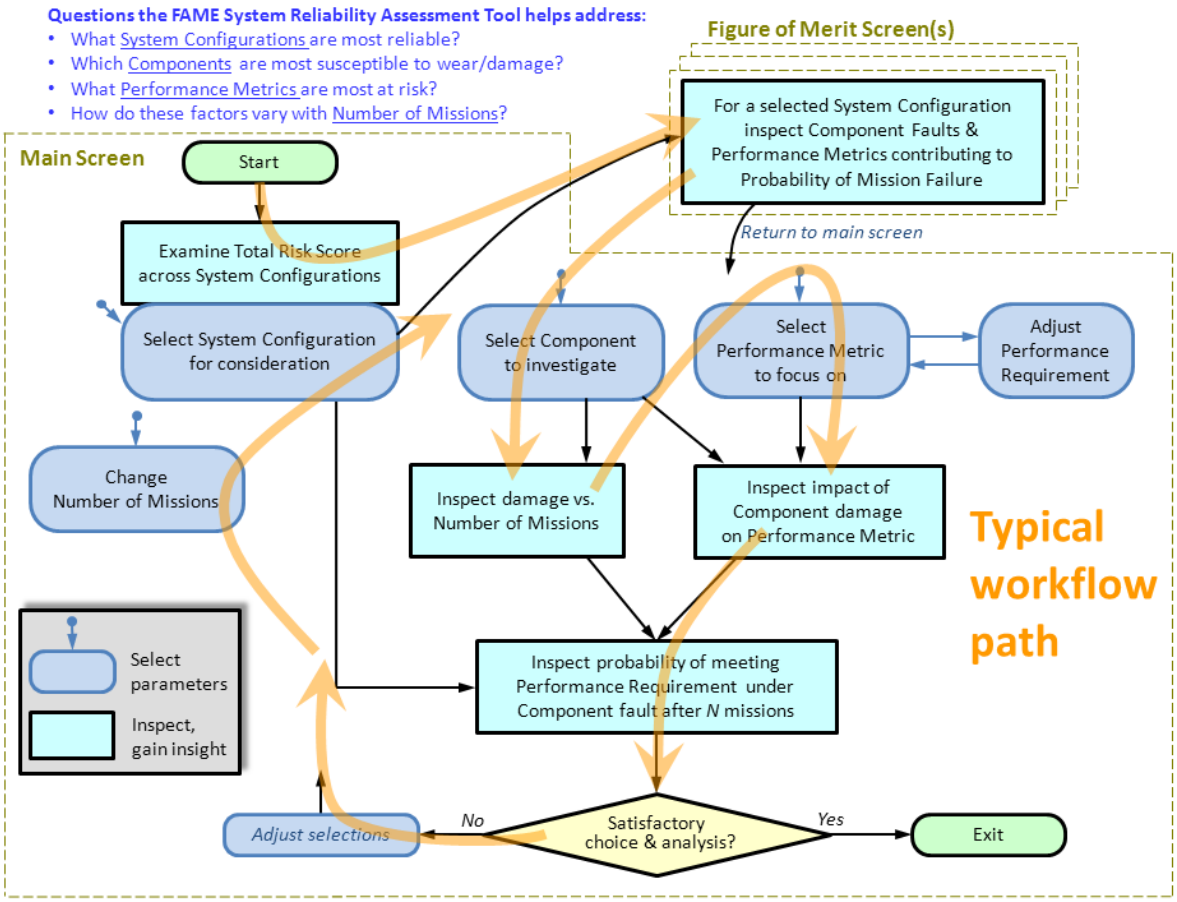

FIGURE 6. TYPICAL WORKFLOW

kph), and a *Required Probability* of meeting the performance metric after the set number of missions. As described above, the probability of meeting the performance metric after damage has accumulated to the set number of missions, is compared with the required probability, to determine system-level probability of mission success on the next mission.

The four graphs presented by the interface are:

- *damage distribution vs. time* as shown in Figure 3.
- *performance vs. damage amount* as shown in Figure 2.
- *reliability vs. time* as shown in Figure 4.
- *probability of mission failure breakdown* as shown in Figure 5.

A conceptual map of the interface and a typical workflow path are shown in Figure 6.

The tool supports two closely related tasks. Task I is to determine the most critical components subject to failure. After selecting a number of missions, the interface displays estimates of overall system reliability in terms of probability of mission success, for each system configuration. From here, the user will typically choose one system configuration to explore in more detail. With a mouse click, they surface a corresponding *probability of mission failure breakdown* graph (Figure 5). Here, they determine which component faults are most responsible for likely mission failure, and on what performance metric(s).

Next, in Task II, the user can explore the critical failure modes more deeply. Guided by Task I, they select a component subject to failure and a performance metric. After a bit of computation by the back-end service, the three Insight graphs are displayed for the user to gain further appreciation about how damage to the selected component evolves over time to adversely impact probability of meeting each performance metric.

Currently, we are demonstrating the capability of the FAME System Requirement Exploration Tool on a fixed system design with six different configurations of components, and a set of thirty-six components subject to damage and failure. The prototype entertains four performance metrics relevant to land ve-

FIGURE 7. GUI FOR FAME RELIABILITY TOOLS

hicle performance. At deployment time, the damage maps have been precompiled, and what is computed dynamically in reponse to user actions is the probability of meeting requirements under user selection of system configuration, number of missions, component subject to fault, and performance metric of interest. In the future, the tool can be extended to novel designs that combine known components in different ways. Thereafter, the tool can be further enhanced to allow designers to customize components, and to input associated (a) failure modes and damage associated with a novel component and (b) Modelica models to represent the functionality of the component. Additional computing hardware will be needed to support timely responses in the event that

damage maps and Modelica simulations will be computed at user interaction time instead of being precomputed as they are now.

CASE STUDY

Explanation of design problem

The FAME based System Reliability Exploration Tool has been implemented for a design of a military vehicle. The corresponding Modelica model for this military vehicle is fairly sophisticated and contains subsystems including Engine, Cross Drive Transmission, Drive Shaft, Final Drive, and Power Take Off Module. Each of these subsystems consists of many components and subcomponents, and parameters for the subsystems are tuned such that they represent known commerical off-the-self (COTS) subsystems. The overall complexity of the model is reflected in over 3200 variables and 10000 parameters. In this case study, a design team could choose from two different Cross Drives and three different Final Drives.[3]

Discussion

Several insights about system reliability and maintenance scheduling can be obtained using the FAME System Reliability Exploration Tool. Three insights are particularly valuable to designers:

1. Determine reliable system configurations.
2. Discover the most critical failure modes for a particular system configuration.
3. Estimate appropriate maintenance schedules.

Because component-wise and performance-metric-wise reliability metrics are aggregated to system level reliability, it is easy to compare reliability performance between system configurations. Figure 8 shows that some system configurations fail at the outset even under nominal, undamaged, component function. Additionally, this vehicle system displays a general trend. System reliablity as function of time usually consists of piece-wise smooth curves. Each smooth segment represents a time interval during which a single fixed set of failure modes contributes most to system reliability, while junctions reflect transitions between different dominant failure modes. Thus, Figure 8 can aid the designer in determining interesting system configurations as well as significant points in mission count. The designer can apprehend that some system configurations such as *Allsion X411-4/Final Drive 3.0* and *Allsion X411-4/Final Drive 2.7* do not meet the requirements even under the nominal design.

A designer can further focus on a particular system configuration and critical operational time. In this case, the designer could look at *Allsion XTG411-A/Final Drive 3.0* at 68 missions

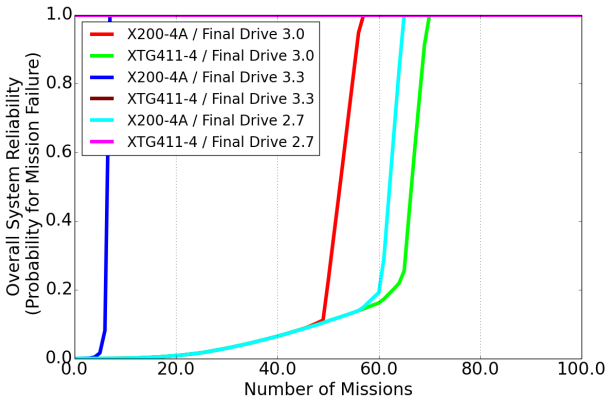

FIGURE 8. SYSTEM RELIABILITY VS NUMBER OF MISSIONS

to determine critical component failure modes (see Figure 9). The designer can determine that *Engine.Inertia.Bearing.Friction* is one of the critical failure modes for this design configuration. Interestly, all of the system configurations use the same engine, but show drastically different impact of engine damage for different combinations of cross drive and final drive. This phenomena would not be captured if reliability were not viewed from a performance metric perspective rather than a component failure perspective.

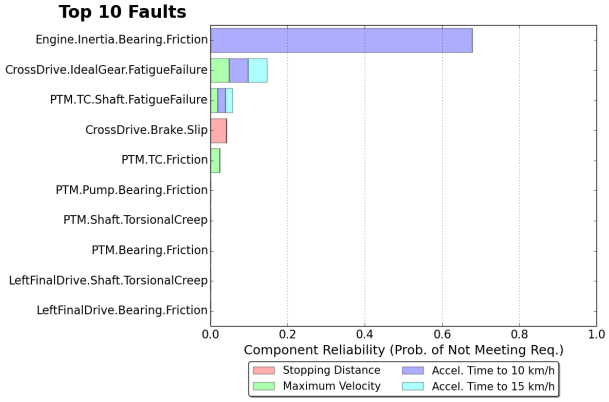

FIGURE 9. EXAMPLE OF CRITICAL COMPONENT FAILURE MODES

Finally, a designer can outline an appropriate maintenance schedule using this tool (see table 1). Maintenance scheduling assumes some acceptable mission failure rate, for example 10%. Interestly, the designer can observe a "water-bed" effect between Engine and PTM for the top two system configurations of Allsion XTG411-A with Final Drive 3.0 and Allison X200-4A with Fi-

[3]We are expecting the model to be released to the public on a web site similar to `http://www.darpa.mil/OpenCatalog/` when the DARPA META project is over.

nal Drive 2.7. The designer will need to choose between slightly longer engine life (61 missions vs 65 missions), and much longer PTM life (82 missions and 96 missions). In practice the designer would take into consideration cost and other factors, but, the outline of a maintenance schedule will inform the overall decisions.

CONCLUSION

This paper has demonstrated the function of a new *System Reliability Exploration Tool* based on the improved simulation capabilities of a system called *Fault-Augmented Model Extension* (FAME). A web-based user interface streamlines workflows to accomplish several goals:

- Evaluate and compare reliability of different system configurations.
- Determine critical failure modes for a particular system configuration.
- Estimate maintenance schedule for a particular system configuration.

The tool has been met with approval in an initial evaluation by reliability experts within the Defense Advanced Research Projects Agency (DARPA) Advanced Vehicle Make (AVM) program, including those from the National Aeronautics and Space Administration (NASA). The tool is currently queued up for evaluation of practicality by a set of professional designers of heavy-duty land vehicles.

ACKNOWLEDGMENT

Material within this technical publication is based upon the work supported by the Defense Advanced Research Projects Agency (DARPA) as part of a subcontract under Vanderbilt University Prime Contract HR0011-13-C-0041. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the official policy or position of the Department of Defense or the U.S. Government.

The authors would like to acknowledge helpful contributions from Gregory Burton, David Garcia, and Nora Boettcher from the Palo Alto Research Center, Michael Lowry and Nathan Wiedenman from DARPA, and Zsolt Lattman and Xenofon Koutosukos from Vanderbilt University.

REFERENCES

[1] de Weck, O. L., 2012. "Feasibility of a 5× speedup in system development due to meta design". In 32nd ASME Computers and Information in Engineering Conference.

[2] Reich, Y., 2010. "My method is better!". *Research in Engineering Design,* ***21***(3), pp. 137–142.

[3] Hazelrigg, G. A., 2010. "Letter to editor re: The pugh controlled convergence method: model-based evaluation and implications for design theory". *Research in Engineering Design,* ***21***(3), pp. 143–144.

[4] Frey, D. D., Herder, P. M., Wijnia, Y., Subrahmanian, E., Katsikopoulos, K., Neufville, R., Oye, K., and Clausing, D. P., 2010. "Research in engineering design: the role of mathematical theory and empirical evidence". *Research in Engineering Design,* ***21***(3), pp. 145–151.

[5] Honda, T., and Antonsson, E. K., 2007. "Grayscale System Reliability: Assessment of Degradation, Reliability, and Lifetime in Engineering Design". In 16th International Conference on Engineering Design (ICED), The Design Society.

[6] Honda, T., and Antonsson, E. K., 2007. "Coupling Effects and Sensitivity Analysis for Grayscale System Reliability". In 19th International Conference on Design Theory and Methodology (DTM), ASME. Paper Number: DETC2007/DTM-35673.

[7] Honda, T., 2007. "Formalization and Applications of Grayscale Reliability Analysis for Engineering Design". PhD thesis, California Institute of Technology, Pasadena, CA, Sept.

[8] Jordan, W., 1972. "Failure Modes, Effect, and Criticality Analysis". In Proceedings of the Annual Reliability and Maintainability Symposium, pp. 30–37.

[9] Greer, D., and Bustard, D. W., 2002. "Collaborative Risk Management". In 2002 IEEE International Conference on Systems, Man, and Cybernetics.

[10] Department of Defense, 1980. *Military Standard: Procedures for Performing a Failure Mode, Effects and Criticality Analysis*, mil-std-1629a ed. Department of Defense, Washington DC.

[11] Veselt, W., Goldberg, F., Roberts, N., and Haals, D., 1981. *Fault Tree Handbook*. U.S. Nuclear Regulatory Commission.

[12] Lee, W. S., Grosh, D. L., Tillman, F. A., and Lie, C. H., 1985. "Fault tree analysis, methods, and applications — a review". *IEEE Transactions on Reliability,* ***R-34***(3), August, pp. 194–203.

[13] Meshkat, L., Girerd, A., and Edwards, C. D., 2006. "An Integrated Approach for the Probabilistic Risk Assessment of the Mars Relay Network". In Annual Reliability and Maintainability Symposium.

[14] Denning, R., 2012. *Applied R&M manual for Defence Systems*. MoD, Abbey Wood, ch. 30.

[15] Pullum, L. L., and Dugan, J. B., July, 1995. Dynamic fault tree analysis methodology examples. Tech. rep., Quality Research Associates, Inc.

[16] Murata, T., 1989. "Petri nets: Properties, analysis and applications". *Proceedings of the IEEE,* ***77***(4), pp. 541–580.

[17] Ciardo, G., Muppala, J., and Trivedi, K., 1989. "SPNP:

TABLE 1. ESTIMATED MAINTENANCE SCHEDULE

COMPONENTS	Allison X200-4A Final Drive 3.0	Allsion XTG411-A Final Drive 3.0	Allison X200-4A Final Drive 3.3	Allison X200-4A Final Drive 2.7
Cross Drive	56	56	12	56
Engine	50	65	38	61
PTM	96	82	86	96
Left Final Drive	>8000	>8000	>8000	>8000
Right Final Drive	>8000	>8000	>8000	>8000
Road Wheel	>8000	>8000	6	>8000

Stochastic Petri net package". In Proc. 3rd International Workshop on Petri Nets and Performance Models, pp. 142–151.

[18] Volovoi, V., 2006. "Stochastic Petri Nets Modeling using SPN". In Annual Reliability and Maintainability Symposium.

[19] Bierbaum, R., Brown, T., and Kerschen, T., 2001. "Model-based reliability analysis". In Reliability and Maintainability Symposium, 2001. Proceedings. Annual, pp. 326–332.

[20] Tiller, M., 2001. *Introduction to Physical Modeling with Modelica*. Kluwer Academic Publishers, Norwell, MA, USA.

[21] Taguchi, G., 1986. *Introduction to Quality Engineering*. Asian Productivity Organization, Unipub, White Plains, NY.

[22] Kackar, R. N., 1985. "Off-line quality control, parameter design, and the Taguchi approach". *Journal of Quality Technology,* ***17***(4), Oct.

[23] Phadke, M., 1989. *Quality Engineering Using Robust Design*. Prentice Hall, Englewood Cliffs, NJ.

[24] Quirante, T., Sebastian, P., and Ledoux, Y., 2013. "A trade-off function to tackle robust design problems in engineering". *Journal of Engineering Design,* ***24***(1), pp. 64–81.

[25] Hoyle, C., Tumer, I. Y., Kurtoglu, T., and Chen, W., 2011. "Multi-stage uncertainty quantification for verifying the correctness of complex system designs". In 37th ASME Design Automation Conference.

[26] Uckun, S., 2011. Meta ii: Formal co-verification of correctness of large-scale cyber-physical systems during design. Tech. rep., Palo Alto Research Center, August.

[27] Aiguier, M., Boulanger, F., Krob, B., and Marchal, C., 2013. "Early stage verification and validation of cyber-physical systems through requirements driven probabilistic certificate of correctness metric". In The 4th International Conference on Complex System Design & Management.

[28] Stone, R. B., Tumer, I., and Wie, M. V., 2005. "The Function-Failure Design Method". *Journal of Mechanical Design,* ***127***(3), May, pp. 397–407.

[29] Feather, M. S., and Cornford, S., 2002. "A Quantitative Risk Model for Early Lifecycle Decision Making". In Conference on Integrated Design and Process Technology, Society for Design Process and Science.

[30] Tumer, I. Y., and Stone, R. B., 2003. "Mapping function to failure mode during component development". *Research in Engineering Design,* ***14***(1), Feb., pp. 25–33.

[31] Meshkat, L., Cornford, S., and Moran, T., 2003. "Risk Based Decision Tool for Space Exploration Misson". In AIAA Space Conference. Paper Number: AIAA 2003-6377.

[32] Feather, M. S., Cornford, S. L., and Moran, K., 2004. "Got Risk? A Risk-Centric Perspective for Space Spacecraft Technology Decision-Making". In Fifth National Symposium on Space Systems Engineering and Risk Management.

[33] Grantham-Lough, K., Stone, R., and Tumer, I. Y., 2007. "The risk in early design method". *Journal of Engineering Design*, Nov.

[34] Kurtoglu, T., and Tumer, I. Y., 2008. "A graph-based fault identification and propagatin framework for functional design of complex system". *Journal of Mechanical Design*(3), Nov., pp. 1–8.

[35] Kurtoglu, T., Tumer, I. Y., and Jensen, D. C., 2010. "A functional failure reasoning methodology for evaluation of conceptual system architectures". *Research in Engineering Design*(4), pp. 209–234.

[36] Tu, J., and Choi, K. K., 1999. "A New Study in Reliability Based Design Optimization". *ASME Journal of Mechanical Design,* ***121***(4), pp. 557–564.

[37] Tu, J., 1999. "Design Potential Concept for Reliability-Based Design Optimization". PhD thesis, University of Iowa.

[38] Youn, B.-D., 2001. "Advances in Reliability-Based Design

Optimization and Probability Analysis". PhD thesis, University of Iowa.
[39] Castanier, M. P., Lamb, D. A., and Mourelatos, Z. P., 2011. "Fleet maintenance simulation for unmanned ground vehicles". In Simulation Based Reliability & Safety, TARDEC.
[40] Shigley, J. E., and Mischke, C. R., 1989. *Mechanical Engineering Design*. McGraw Hill.
[41] Norton, R. L., 2000. *Machine Design: An Integrated Approach*. Prentice Hall.
[42] Jones, M. H., and Scott, D., 1983. *Industrial Tribology: The Practical Aspects of Friction, Lubrication and Wear*, Vol. 8. Elsevier.
[43] Harris, T. A., and Kotzalas, M. N., 2006. *Advanced concepts of bearing technology: rolling bearing analysis*. CRC Press.
[44] Singhal, S., 2008. "Sleeve bearing design for slow speed applications in cement plants". In Cement Industry Technical Conference Record, 2008 IEEE, IEEE, pp. 283–290.
[45] Maru, M. M., and Tanaka, D. K., 2007. "Consideration of stribeck diagram parameters in the investigation on wear and friction behavior in lubricated sliding". *Journal of the Brazilian Society of Mechanical Sciences and Engineering,* ***29***(1), pp. 55–62.
[46] de Kleer, J., Janssen, B., Bobrow, D., Kurtoglu, T., Saha, B., Moore, N., and Sutharshana, S., 2013. "Fault Augmented Modelica Models". In The 24th International Workshop on Principles of Diagnosis.
[47] Makarichev, A. V., 1995. "Estimates of system failure probability during the renewal time for a group of repairable systems". *Cybernetics and System Analysis,* ***31***(6), Nov-Dec, pp. 931–934.