System Qualities (SQs) Ontology, Tradespace and Affordability (SQOTA), Phase 6: 2017-2018

Technical Report SERC-2018-TR-108

June 21, 2018

Principal Investigator:

Principal Investigator: Dr. Barry Boehm, University of Southern California

Research Team:

Air Force Institute of Technology

Georgia Institute of Technology

Massachusetts Institute of Technology

Naval Postgraduate School

Pennsylvania State University

University of Southern California

University of Virginia

Wayne State University

Sponsor: DASD(SE)

The Systems Engineering Research Center (SERC) is a federally funded University Affiliated Research Center managed by Stevens Institute of Technology.

This material is based upon work supported, in whole or in part, by the U.S. Department of Defense through the Office of the Assistant Secretary of Defense for Research and Engineering (ASD(R&E)) under Contract HQ0034-13-D-0004, TO # 0283.

Any views, opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Department of Defense nor ASD(R&E).

TABLE OF CONTENTS

Table of Contents iii

List of Figures iii

List of (Tables, Sequences) iv

Executive Summary 1

AFIT RT-181 Final Report 2
- **SysML Reference Architecture** 2
- **Architecture Based Evaluation of Space Concepts** 3

GTRI RT-181 Final Report 5
- **Background** 5
- **Objectives of Prior Work** 6
- **Phase 5 Activity** 7
  - Activity 1: Tradespace Methods, Processes, and Tools (MPTs) 7
  - Activity 2: Next Generation Cost Modeling 16

MIT RT-181 Final Report 19
- **Ilities Semantic Basis** 19
  - Prior Phase Research Background 19
- **Phase 6 Accomplishments** 21
  - Metrics Derived from the Semantic Basis 23
  - Potential Opportunities for Applying the Semantic Basis 26
  - Extending the Basis 28

NPS RT-181 Final Report 29

PSU RT-181 Final Report 30

USC RT-181 Final Report 30

U. Virginia RT-181 Final Report 31

Wayne State RT-181 Final Report 31

Appendix A: List of Publications Resulted 42

Appendix B: Cited and Related References 43

LIST OF FIGURES

Figure 1 Tradespace Analysis "Tap Points" 4

Figure 2 GTRI's objective in support of ITAP with a focus on MPTs to support analytical foundations and next-generation cost estimation models 5

Figure 3 Example expert basic probability assignments for scenario input variables 11

Figure 4 System 1 response variable 'intercept rate' evaluated with respect to belief and plausibility that it will exceed a Required Intercept Rate 12

Figure 5 System 2 response variable 'intercept rate' evaluated with respect to belief and plausibility that it will exceed a Required Intercept Rate reflects the overall better performance compared to System 1 13

Figure 6 System 2 response variable 'intercept rate' evaluated with respect to belief and plausibility that it will exceed a Required Intercept Rate 14

Figure 7 System response variable 'intercept rate' evaluated with respect to belief that it will exceed a Required Intercept Rate for two different system designs 15

Figure 8 Welcome screen with ilities dictionary choice 20

Figure 9 Constructed ilities statement 20

Figure 10 Welcome screen with ilities dictionary choice 21

Figure 11 Elements in the Ilities Semantic Translation Layer Assistant prototype 22

Figure 12 Core Supporting Constructs for the Ilities Semantic Translation Layer Assistant 22

Figure 13 Semantic basis as supplying information for antecedent description, state counting, and path valuation 23

Figure 14 Example ility label relation to existence, degree, and value of state changes 24

Figure 15 Example ility label relation to existence, degree, and value of state changes 24

Figure 16 Example selective filtering of outdegree 25

Figure 17 Example use of multi-epoch analysis to quantify robustness 25

Figure 18 USC ility term (initial draft) mapping to basis 28

Figure 19 Comparison of USC (initial draft) to MIT ility term label mapping 29

LIST OF (TABLES, SEQUENCES)

Table 1 Design parameters for two different notional BMD systems 12

EXECUTIVE SUMMARY

During Phase 6, the Systems Qualities Ontology, Tradespace, and Affordability (SQOTA) team has made significant progress, not only in research results, but also in their application and support of DoD organizations and FFRDCs. Several team members are addressing coordinated control of multiple autonomous vehicles: Wayne State with TARDEC for ground vehicles; Penn State with the Navy for sea vehicles; AFIT and NPS for Air Force and Navy swarms of airborne drones, also in collaboration with USC's Azad Madni's Hidden Markov Model-based machine learning capability for dealing with uncertainties in the control of swarms of airborne drones. Georgia Tech continues to extend its versatile model-based systems engineering toolset in support of Army, Navy, and Marine Corps systems engineering and tradespace analysis efforts, and is extending its capabilities to address security tradespace analysis.

USC's Systems Qualities Ontology is being used to enable tradespace analysis between other system qualities (SQs) and Maintainability, which supports not only Total Ownership Cost (TOC) and Changeability, but also Dependability in terms of reducing Mean Time To Repair. USC is also developing, applying, evaluating, and extending a big-data analysis toolset called Systems Qualities Understanding through Analysis of Abundant Data (SQUAAD), which has been used to analyze a Navy Safety tradespace analysis application. USC is also coordinating its Maintainability ontology with MIT's Changeability semantic ontology, and with U.Virginia's big-data analysis capabilities. The USC Ontology has also been used to clarify the quagmire of alternative definitions of Resilience, in concert with Aerospace Corp's March 2018 Ground Systems Architecture Workshop, and is preparing to extend its SQUAAD capabilities to address Security and Interoperability tradespace analysis.

In the next-generation cost modeling area, Phase 6 included a collaboration of NPS and USC with the Naval Center for Cost Analysis in addressing early cost analysis of agile systems developments. USC has also completed an initial calibration of the COSYSMO 3.0 upgrade in a team effort with NPS and several aerospace companies, including effects of systems engineering artifact reuse, and is exploring cost estimation of security levels in concert with the Software Engineering Institute.

This report summarizes and provides examples of the overall SQOTA Phase 6 results by team member organizations in alphabetical order.

AFIT RT-181 FINAL REPORT

In SQOTA Phase 6 research, AFIT pursued two primary streams of research related to early cost, effectiveness and "ility" analysis. The first stream continued on the previous year's work with architectural analysis applied to Multi-UAV concepts. New developments this past year involved the development of a reference architecture for small UAS (sUAS) constructed in the Model-Based SE (MBSE) tool Cameo Systems Modeler. Further, a concept application of a sUAS based Remote Targeting System (RTS) was modeled functionally and physically using the reference architecture. AFIT worked with the Naval Postgraduate School (NPS) to demonstrate an approach to parsing the architecture for COSYSMO cost drivers to produce relative cost estimates, demonstrating key extensions to current MBSE approaches. The second stream of research expanded AFIT's architectural based cost effectiveness analysis to a new domain – that of space based remote sensing. AFIT developed an architecture and effectiveness analysis model comparing a CubeSat constellation to a traditional monolithic satellite architecture with a scenario based on hurricane damage assessment associated with the recent Hurricane Maria landfall in Puerto Rico. SysML-based cost analyses for the satellite variants were developed from the reference architectures by NPS. Preliminary cost estimates of CubeSat vs. traditional satellites were based on COSYSMO size drivers, yet other sources of cost variance for hardware were not captured. This is not completely surprising since COSYSMO's roots are for software systems, and this will be addressed in Phase 7 to better reflect expected cost differences. These past year's research will be discussed in more detail subsequently.

SYSML REFERENCE ARCHITECTURE

Several SysML MBSE tools have become full featured modeling environments, moving beyond static architecture depictions to now include parametric trade space analysis, verification of design requirements, and evaluation of behavior and performance of the design. This enables early evaluation of a concept based on an architectural definition, well in advance of a detailed system design. AFIT has been exploring the use and reuse of MBSE architectures based on SysML. A reference architecture for sUAS has been developed in Cameo Systems Modeler, a SysML MBSE tool developed by NoMagic, Inc. The reference architecture includes a library of predefined components for common sUAS components, defined in terms of technical parameters, interfaces and behaviors. The component library includes air vehicle, ground station and mission payload components. Parametric diagrams allow evaluation of common performance metrics such as vehicle range, mission endurance, communication range, image ground resolution, and area coverage rate. sUAS concepts can be evaluated against mission needs by developing Use Case models and Activity models in the tool to determine functional requirements for the concept. Physical architectures can then be built from the library components, with subsequent evaluation using the parametric diagrams. Multiple models can be constructed with different choices of components and/or different parameter values, thus facilitating trade space analysis.

There are several evaluation efforts associated with AFIT's sUAS Reference Architecture. An initial effort was conducted based on a Remote Targeting System (RTS) concept previously considered by AFIT and NPS. Prior evaluation of the RTS concept was done using a non-SysML tool for performance analysis, and cost parameters were exported to COSYSMO for subsequent evaluation by NPS researchers. Results of this analysis were presented in November, 2017 at the SERC Annual Review. Since then, the RTS concept was re-created using the sUAS Reference Architecture, and a method for direct machine parsing of the architecture for the cost parameters was developed.

A second application of the sUAS Reference Architecture involved evaluation of vulnerability of sUAS. An MS student from the National Air and Space Intelligence Center (NASIC) used the architecture identify possible avenues of approach for performing counter UAS activities. sUAS architectures commonly used by rogue operators and lower-tier adversaries were defined using the library components in the reference architecture. These architectures included fully autonomous vehicles (no communication between vehicle and ground station), semi-autonomous control, and flight conducted via First Person Video (FPV). Attributes of the system were identified based on their ability to impact detection, tracking/orientation, and or direct attack options. The analysis will be used to define counter-UAS strategies, and will be validated with upcoming flight tests. This research was documented in a limited distribution thesis, published through the Defense Technical Information Center (DTIC).

A third application of the sUAS Reference Architecture is ongoing with a class project. AFIT offers a three course graduate specialization covering sUAS design, development and evaluation. The courses utilizes a project based approach, sponsored by another AF organization, whereby a mission problem is given to the student groups, requiring them to execute a rapid prototyping process to conceptualize, design, build and flight test an sUAS concept to address the mission need. As part of the project, the students were provided with the sUAS Reference Architecture, and asked to use that to develop their design, provide requirements traceability, and facilitate verification and validation of their approach. Students have been using the parametric diagrams to perform tradespace analysis. While the project is ongoing (completion in August, 2018), use of the sUAS Reference Architecture has improved architecture quality over past years, and students have indicated that it saves cycle time by allowing them to rapidly evaluate candidate vehicle designs.

ARCHITECTURE BASED EVALUATION OF SPACE CONCEPTS

Historically, operational space missions have been performed by large, highly-capable satellite systems. While these traditional satellite designs meet or exceed very demanding performance and reliability requirements, their drawbacks of expense and potential vulnerability have incentivized consideration of alternative architecture types. The trade space for alternative architectures is limited only by creativity; architectures for a given mission could vary by satellite size, number, orbit, payload types, design life, or any other number of parameters or concepts. Evaluation of concepts and options within this trade space is helped by a structured and systematic approach; Model-Based Systems Engineering provides such an approach. This research explored the practical usage of Model-Based Systems Engineering to assess one possible alternative to traditional satellite architectures. Since their inception in 1999, CubeSats have been used for education and technology demonstration. More recently, commercial companies have begun to deploy constellations of CubeSats for Earth imaging and other purposes. CubeSats have the potential for being both cheaper and potentially less vulnerable, though with sacrifices in capability or performance. As an example of how MBSE can be used to assess solutions from a vast space architecture design trade space, this research developed static and dynamic architectural models using MBSE and SysML comparing traditional and CubeSat architecture performance in a disaster response imagery scenario. The scenario modeled was based on the Hurricane Maria landfall over Puerto Rico in 2017.

The model developed for the traditional architecture, and its derivative CubeSat model, were used for one limited application. The figure below illustrates where “tap points” for varying trade space exist in the four steps of the OOSEM System Specification and Design Process. It would be relatively straightforward

to incorporate SysML models of other design solutions for this scenario, such as UAVs or a mixture of different sensors and satellites. Specific to the CubeSat/Traditional architecture comparison, this research involved performance evaluation and relative cost estimation. The performance evaluation demonstrated the ability to meet threshold requirements using the CubeSat architecture, but fundamental limitations associated with small aperture cameras on the CubeSats did not allow it to meet objectives associated with ground resolution. The cost estimation effort, performed by NPS researchers, utilized our previous approach to parsing the architectures for cost parameters (scenarios, requirements, blocks/algorithms and interfaces) used as inputs to COSYSMO. Results from the COSYSMO analysis were inconclusive for this initial investigation as they did not reflect major differences in the cost estimates based on the two very different system scales considered; the sheer size and complexity differences associated with traditional and CubeSat architectures. This indicates the need for additional Cost Estimating Relationships (CERs) associated with the size of the physical system. These CERs are available specific to the domain of application, but they are not fully incorporated within COSYSMO. This highlights an important area of research to be pursued in the coming years. This research was documented in an AFIT MS Thesis and was presented at the Conference on Systems Engineering Research (CSER) in May 2018, with subsequent publication in the proceedings for the conference.

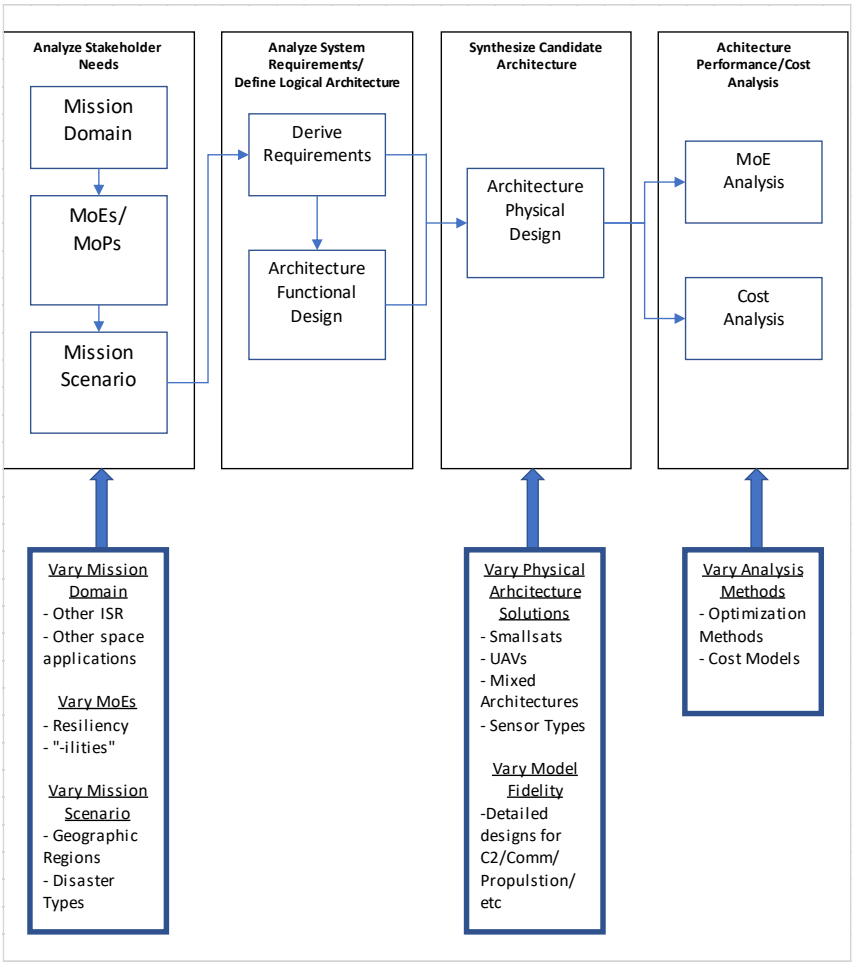


Figure 1 Tradespace Analysis "Tap Points"

BACKGROUND

This effort is proposed in support of the System Qualities Ontology, Tradespace and Affordability (SQOTA), originally called ilities Tradespace and Affordability Project (iTAP), executed through the Systems Engineering Research Center (SERC). Its main objective is to provide DoD-community systems engineers with stronger foundations and methods, models, processes, and tools (MMPTs) for dealing with the complex and system-critical interactions among a system's quality attributes (SQs), also called ilities or non-functional requirements (NFRs). The SQs are weakly and inconsistently defined, often underemphasized in DoD acquisition reviews and guidance, and often the major source of system acquisition and support shortfalls and overruns.

The SQOTA effort has been executed in several phases to date. GTRI's efforts have focused on the development of methods, processes, and tools necessary to realize executable evaluation of the SQs and integrating system engineering cost models developed by SQOTA collaborators (University of Southern California and Naval Postgraduate School) with model-based systems engineering frameworks such as SysML and OpenMBEE. A diagrammatic view of GTRI's objectives in support of SQOTA is presented in Figure 2.

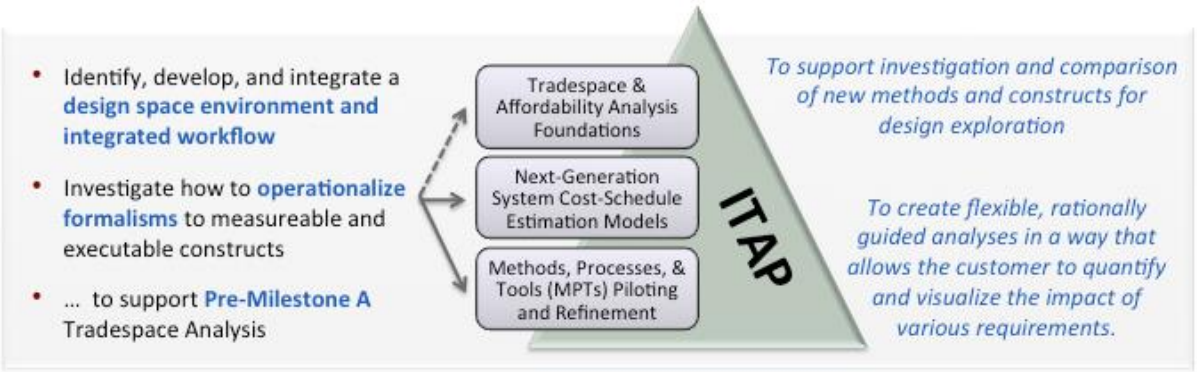


Figure 2 GTRI's objective in support of ITAP with a focus on MPTs to support analytical foundations and next-generation cost estimation models

GTRI's work in Phase 1 (January – May 2013) set the foundation and framework and demonstrated initial proofs-of-concepts based on SERC team member's existing capabilities. The Phase 2 activity (May – December 2013) improved and piloted several existing ITAP analysis toolsets based on the results of Phase 1. Phase 3 activities (January – December 2014) extended this work to include deployment of an integrated proof-of-concept toolset for tradespace analysis. Phase 4 activities investigated maturation of modeling concepts in support of systems engineering analyses that would result in more effective and efficient tradespace exploration in a computational environment. Phase 5 activities focused on development of a framework that would enable systems engineers (SEs) to answer unique questions, architecting the necessary analysis with respect to regions of a tradespace to evaluate and the order in which various methods are applied.

This document reports on GTRI's Phase 6 activities (22 June 2017 – 13 June 2018), which sought to further mature and investigate new pathways with respect to methods, tools, and processes

investigated earlier in ways that increase relevance to operational analysis needs in support of the DoD acquisitions process.

OBJECTIVES OF PRIOR WORK

The primary objective of GTRI's prior effort was to develop an integrated workflow process to guide design exploration. Further, this workflow was designed to explicitly consider how the context in which a system is used influences its overall value to stakeholders and, importantly, across simultaneously competing or sequentially changing needs of these stakeholders. As it is used here, context can refer to how the additive value of a system varies between stakeholders, or temporal differences in a system's application over its lifecycle that impact its perceived usefulness. Further efforts investigated how to facilitate modeling in support of tradespace generation and analysis to capture specific operational scenario needs, hence increasing operational relevance of these approaches.

As an initial proof of concept an open source, web-based toolset was developed to couple a rationally guided workflow to existing analysis methods for design tradeoff evaluation. This toolset leverages existing open source web frameworks such as the Django web application framework, and the D3 Javascript visualization libraries to enable the rapid development of a complex, database-driven website. NASA's OpenMDAO framework, another open source tool, is used to facilitate complex analysis by linking together the separate models used to describe the behavior and performance of the system of interest. In addition to the use of several open source software frameworks, the toolset described here leverages a SysML modeling language inspired data model to specify the parametric constraints that define system performance.

Another proof of concept, the PAW framework, extended the analytical abilities of SEs to address questions unique to a given problem or system, especially with respect to operational context. PAW builds off the previous work and provides modularity and repeatability in analysis workflows for systems engineering and a common framework for data handling in analysis (e.g. converting tradespace matrices or stochastic data into json). The Python package Luigi and Pandas library provide the core mechanism for pipelining component execution and basis for structuring and passing data through the framework, respectively.

Other work integrating SysML and existing systems engineering cost models (COSYSMO) with collaborators USC and NPS worked on enhancing and refining the generic SysML-based cost modeling building blocks. These building blocks and their underlying cost modeling principles are generic and thus can be applied to practically any system. Through various case studies, this portion of the effort has also been exploring additional technologies that can make interacting with these models more intuitive and coinvestigating the potential of OpenMBEE to help fulfill the model interoperability framework vision. OpenMBEE is a SysML-based modeling and simulation environment that is being used in production on several major NASA projects. It has been developed and open-sourced by NASA JPL, and may be considered as a flexible "model-based wiki" frontend for the cost modeling building blocks. The envisioned ultimate result is an easy-to-use web browser frontend that provides the rich SysML-based backend capabilities to a general audience. These efforts have been part of an ongoing collaboration with SQOTA colleagues at the University of Southern California and Naval Postgraduate School to move the previous SysML/COSYSMO integration work toward an easy-to-use web browser frontend that provides the rich SysML-based backend capabilities to a general audience (i.e., they can take advantage of SysML without having to learn SysML.

PHASE 5 ACTIVITY

Phase 6 builds on the research already conducted to date. Phase 6 activities continued to pilot the development, refinement, and application of SERC methods, processes, and tools to DoD-system SQ tradespace and affordability issues in support of:

- SQOTA Task 2: SQ Methods, Processes, and Tools (MPTs) Piloting and Refinement, and
- SQOTA Task 3: Next-Generation, Full-Coverage Cost Estimation Model Ensembles

These are described below as GTRI Activity 1 or 2, respectively.

ACTIVITY 1: TRADESPACE METHODS, PROCESSES, AND TOOLS (MPTS)

Overview

For Phase 6, GTRI focused on methods and processes that could be employed within the previously piloted frameworks and tools as well as capture different aspects of design and synthesis with operational context that would be highly relevant to DoD materiel design and development. Specifically, the effort sought to address contextual and other non-simple sources of uncertainty in tradespace analysis.
The underlying philosophy of this study is to account for uncertainty in the system and the context of use because to neglect these aspects can result in a system-use model that is too simple and consequently gives a false sense of clarity and confidence in the performance attributed to a given system design. Without accounting for this uncertainty earlier in the design process, the resulting system specifications will frequently require repeated modification and changes to account for deficiencies discovered later during testing when uncertain disruptive factors are encountered.

Uncertainty

From a high-level systems viewpoint, uncertainty represents limited knowledge or confidence in the level of predictability of a system state under specific circumstances. Uncertainty may arise due to a lack of information, too much information, conflicting information or evidence, ambiguity (being uncertain about the uncertainty), error in modeling or measurement (i.e., because not all the parameters of the system and their interactions are fully known), and belief [Zimmermann, 2000]. In systems engineering, there are typically three types of recognized uncertainty:

Aleatory uncertainty, which describes the inherent variation of the system, may be modeled mathematically via probability theory. This uncertainty is irreducible regardless of additional efforts put forth by decision makers.

Epistemic uncertainty results from a lack of knowledge or data and is reducible. When epistemic uncertainty refers to variability and uncertainty itself, it is termed "uncertainty about uncertainty" or second-order uncertainty [Einhorn and Hogarth, 1985].

Error, which is treated as a type of uncertainty but one not due to a lack of knowledge.

One of the most critical aspects of effective system engineering, especially during the system design and performance characterization phases, is the identification and quantification of the various sources of

uncertainty that can impact the understanding and characterization of the predicted system state or performance associated with a given design.

There are many different ways to account for and analyze system design and performance in the face of uncertainty. Many of these methods apply uncertainty to the input design variables in the form of a distribution that is then propagated through the system performance models to produce a related distribution of output parameters. Other investigations have focused on uncertainty associated with the decision-making process where the ranking of stakeholder preferences or how the system performance attributes are valued by those stakeholders is not completely known with confidence. The ranking and how attributes are valued correspond to attribute weights and value functions in commonly applied multi-attribute value (MAV) decision methods. Previous work by GTRI [Sitterle et al., 2016] investigated the impact of uncertainty in the weights and value functions in standard MAV decision methods as well as a special variant derived for defense applications where a penalty is ascribed if a key attribute fails to meet a stakeholder-defined threshold. The work illustrated how valuation uncertainty can result in highly skewed or discontinuous distributions, and uncertainty on weights and values can produce a bimodal aggregate distribution when an operational penalty is applied. These analyses emphasized that simply taking a mean, standard deviation, or quartile representation – as is typical practice – may not represent the uncertainty well. Uncertainty analysis concerning identifying a "best set" of alternatives should focus on what aspects of uncertainty change our decision about which design alternatives to include in that set.

Uncertainty is typically measured using probabilistic methods. However in many types of systems engineering models, the modeled abstraction of the problem is inherently conceptual, and probabilities are difficult if not impossible to accurately describe. In these cases, inputs to a model may be known to exist within "reasonable" intervals, but nothing about the mean, variance, or even the type of distribution is known [Argarwal et al., 2004]. In such cases, subjective possibilities are more representative and easier to intuitively describe than frequentist probabilities. There is often no way to generate even a small set of meaningful experiments with which to generate a frequency estimate. Similarly, it is not always possible to infer a frequency estimate from prior events. This is especially true for defense and weapons system evaluation problems where past events are based on conditions dissimilar from current and future analytical contexts.

In the absence of information, for example, many methods make assumptions about the nature of the input data, specifically reflecting these assumptions in how the variables are modeled. Uncertainty is often modeled as uniformly or normally distributed and errors are assumed to be independent. Real data may not fit these assumptions, however, and a method accounting for uncertainty that allows for ignorance may offer better insight with respect to system performance [Foley, 2012].

One approach to dealing with uncertainty in these contexts that has been widely applied to decision fusion based on data from multiple sensors is Dempster-Shafer theory. Dempster-Shafer theory (DST) is a form of evidential belief reasoning devised as a means of dealing with imprecise evidence and potentially conflicting beliefs. At a high level, DST introduces the notion of assigning beliefs and plausibilities to possible measurement hypotheses along with required combination rules to fuse them [Abdallah et al., 2013; Argawal et al., 2004; Foley, 2012; Wu et al., 2002]. The Dempster-Shafer approach is often considered to be a generalized Bayesian theory. In contrast to Bayesian inference, however, *a priori* probabilities are not required in Dempster-Shafer inference because they are assigned the instant the information is provided.

Evidence is represented in a non-negative set function called a mass assignment, *m*, that models a range of possible beliefs about a propositional hypothesis. A mass assignment *m*(U), also called a basic probability assignment (BPA), intuitively describes the extent to which the evidence supports U. For example, assume adversaries use methods P and Q to attack a given system. An expert engineer believes with a 70% degree of certainty that adversaries will attack a given system using "Method Q" and beyond

that does not know. The mass functions will be defined as m(Q) = 0.7 and m(all) = 0.3 (notably, not m(P) = 0.3). This is a key distinction from frequentist methods. DST probability assignments distribute the remaining belief over the universal hypothesis, whereas classical probability distributions distribute it over the complement of the current hypothesis.

Evidence does not assume a particular value or even likelihood of any value in favor of any other within an interval associated with a BPA. For example, the set "Method Q" above might be represented by the interval [42, 57]. DST assumes no preference for any values within that interval over another, just the belief that evidence supports that the true value is within those bounds to some specified degree of certainty. Because of this uncertainty in the information, the BPA for a given event and the BPA for its negation do not need to sum to unity.

Each sensor in the case of sensor fusion, or expert in the case of expert elicitation defining input parameters to a design scenario, will assign its/his/her beliefs over the frame of discernment, or the set of all possible mutually exclusive context facts or events. Belief of an event or scenario is calculated by summing the mass assignments (or BPAs) of the propositions that completely agree with that event or scenario. Belief accounts for all evidence that supports a given proposition and is effectively a lower bound of the DST confidence interval. In contrast, plausibility of an event or scenario is calculated by summing the mass assignments (or BPAs) of the propositions that completely agree and partially agree with that event or scenario. Plausibility accounts for all observations that do not rule out the given proposition and is therefore an upper bound on that confidence interval.

Evidence from different sources are aggregated using rules of combination, and DST assumes these sources are independent. For each possible proposition, DST gives a rule for combining the evidence from the multiple sources that derives a common shared belief between the sources and ignores all the conflicting (non-shared) belief through a normalization factor. The DST combination rule is appropriate when the differences expressed by sources may be viewed as preferences or beliefs on a spectrum. (Issues relevant to cases where there is little to no consistency among the sources, especially when the sources express differences of opinion about events or scenarios that are "either/ or" in nature, have been described by many including [Zadeh, 1986] and are discussed in the 'Summary and Insights' section.)

The outputs of the DST process are a quantification of the beliefs and plausibilities for all possible propositions described that allow one to reason about the degree of certainty or uncertainty of the beliefs. Namely, a small difference between belief and plausibility shows a fairly strong certainty about a belief, while a large difference shows greater uncertainty about that belief. DST consequently does not produce a definitive "answer" (i.e., "pick this design"). It may, however, provide insights lacking by the rote application of multi-additive value methods based on utility theory.

Sample Problem

This work sought to identify a sample problem trivial enough to focus on the incorporation of uncertainty methods instead of a highly detailed system design but rich enough to provide insightful and meaningful results. The system model selected is derived from the ballistic missile defense (BMD) example presented in [Wilkening, 2000]. The model is probabilistic in nature and specified not at a detailed first-principles level but instead one that can be used to determine the technical performance required for a defense to meet specific objectives. The defensive objective is defined as a certain probability that no warheads leak through the BMD system. The technical performance is captured by the interceptor single-shot probability of kill and the warhead detection, tracking, and classification probability. Attacks are characterized by the number of warheads and un-discriminated decoys.

In keeping with the nature of many real systems undergoing analysis in defense contexts, a non-deterministic modeling approach was used in this work. Specifically, randomness is involved in creating the model outcomes; consequently, outputs cannot be explicitly predicted. This mirrors the non-

deterministic engineering system design problem described by [Agarwal et al. 2004] where system responses may be non-unique due to the existence of uncertainties in the input parameters of the model, or there may be alternative mathematical models for a system and its environment.
The BMD model adapted for this effort is essentially a complex Markov Chain for modeling terminal phase defenses with adversary spoofing capabilities included. It contains the following:

Adversary Systems

A warhead object represents a single terminal phase warhead that is either a decoy or real. The warhead object only knows how long it takes to get to a target and whether or not it's a decoy.

An adversary represents a system with 2 active processes trying to:

1. Jam the radars that detect incoming terminal phase warheads, and
2. Jam the command and control (C2) systems to prevent them from signaling launches.

There are probabilities associated with each of these processes.

BMD Systems

(The BMD system consists of radar, command and control, launcher, and interceptor missile elements.)

A radar object represents a single radar system for detecting terminal phase warheads. A radar detects individual warheads and keeps a list of them for the C2 system to evaluate if they are a decoy or real. Radars can be jammed by the adversary and also have probabilities associated with their processes (e.g., probability of detecting a warhead).

A command and control (C2) system object actively evaluates warheads identified by all the radars assigned to it. If a warhead is assessed to be a threat, the C2 system signals a launcher to launch an interceptor missile. The C2 system waits until the interceptor has a non-pending status (i.e., failed) before launching another. When jammed by the adversary, a C2 system will not be able to signal launches.

The interceptor object is launched to target a warhead. The interceptor must acquire the warhead and then has some probability of successful interception.

A launcher object receives signals from a C2 system to launch an interceptor.

The BMD model and all associated analyses were developed in Python and the stochastic simulation of the BMD performance used SimPy, a process-based discrete-event simulation framework based on standard Python.

Approach

The goal for this effort centered on how to choose a “best design” based on highly uncertain evidence. To capture uncertainty in inputs to a system model and also in the model (and its simulation) respectively, the approach focuses on uncertainty in the operational scenarios a notional BMD system might face as well as uncertainty associated with the technological and system performance. The former is defined as inputs based on expert elicitation while the inherent uncertainty associated with the latter is modeled via stochasticity and non-deterministic system behavior.
In the adaptation created for this effort, simulated “experts” provide basic probability assignments (BPAs) over a set of intervals for each uncertain input variable. These BPAs correspond to the probability with

which the actual value will fall within the given interval. This also assumes that the system behavior is roughly monotonic within the intervals defined, which may affect the intervals selected. An example is presented in Figure 3. In this example, the expert determines that the 'decoy_ratio' has a 0.5 probability of falling within the interval [0, 0.3].

```
# rogue state
expert_1 = dict(n_warheads={(2, 6): 0.5, (6, 12): 0.4, (12, 20): 0.1},
          decoy_ratio={(0.0, 0.3): 0.5, (0.3, 0.6): 0.5},
          p_jam_radar={(0.0, 0.1): 0.7, (0.1, 0.3): 0.3},
          tmax_jam_radar={(0, 3): 0.6, (3, 7): 0.3, (7, 8): 0.1},
          nmax_jam_radar={(0, 2): 0.5, (2, 5): 0.5}
         )
```

Figure 3 Example expert basic probability assignments for scenario input variables

In the case of cyber-physical system security, this problem is analogous to representing some potential unknown attacks to the system based on expert elicitation. It is probably too difficult and overly complicated to ask experts to try and come up with true probabilistic measures of these types of inputs, and there are many elicitation approaches that could augment and/or refine how the BPAs are determined. This effort is instead focused on the uncertainty associated with these BPAs on the inputs. With an increasing number of input states and associated BPA intervals assigned for each that are defined, more will be known about uncertainty of the system behavior. However, this will also increase the computational burden associated with the analysis.

The general procedure used by [Argarwal et al., 2004] is incorporated in this effort. Dempster's rule of combination is used to combine the beliefs of the experts with respect to the operational scenario the notional BMD system might face. Using the combined beliefs – evidence in DST – all possible sets of scenario intervals and their BPAs are obtained. From these sets, a list of points to simulate is created. This list is based on the combinations of interval values, and care is taken not to simulate the same point twice as some of these points are the same for different interval combinations.

Each interval set is evaluated to obtain upper and lower confidence intervals associated with the BMD system response variables. System response variables are assumed to be monotonic in the range for each interval. Notably, these are not upper and lower bounds as calculated in [Argarwal et al., 2004] but intervals. This is because while only a single, fixed system was used as the basis of this analysis, it was modeled as a non-deterministic system. ([Argarwal et al., 2004] used a deterministic system model.) Then, belief and plausibility that BMD system response variables exceed or equal their respective performance requirement, e.g., $y_i \geq y_{i\_required}$, are calculated.

Results

As an example of how to use this type of approach, assume that the key stakeholder has set a Required Intercept Rate (RIR) for the notional BMD system. The system response variables for intercept ratio output (as distributions) by the model simulation, can be combined to produce plots of the confidence intervals for belief and plausibility of meeting the RIR in the face of uncertainty associated with (a) the operational scenario the BMD system will face as characterized by the experts, and (b) the technical system performance itself as captured via a non-deterministic simulation approach.

In this case, the set of interest is the interval $y \geq RIR$; sets that intercept this are combinations of input variables where the minimum of the *intercept_ratio_statistic is* $\geq RIR$. Plausibility is defined as the sum of all the masses of the sets that intersect the set of interest. Here again, the set of interest is the interval $x \geq RIR$; sets that intercept this are combinations of input variables where the maximum of the *intercept_ratio_statistic is* $\geq RIR$.

In Figure 4, the belief and plausibility that the intercept rate is greater than or equal to the Required intercept Rate (RIR) is plotted as a function of RIR. The graph reveals, for example, that the 95% confidence interval belief that the notional BMD system will meet an RIR of 0.4 is between 0 and 0.2.

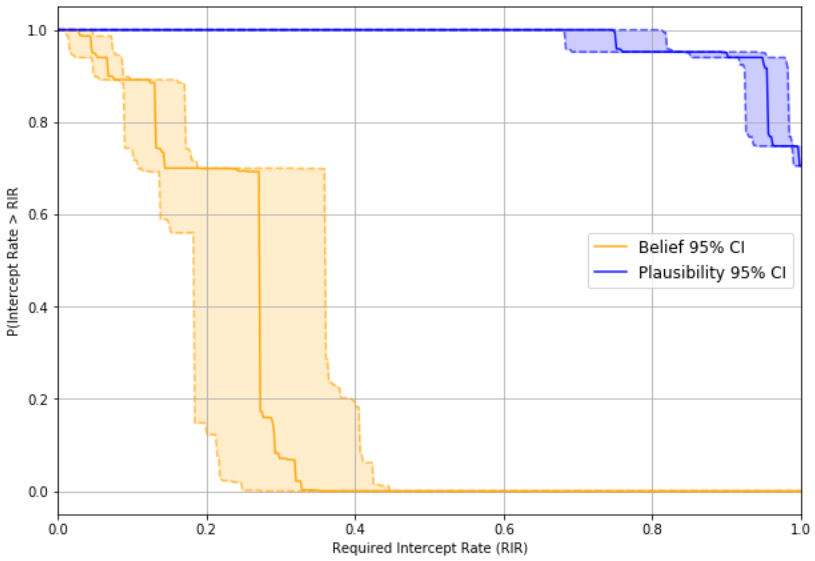


Figure 4 System 1 response variable 'intercept rate' evaluated with respect to belief and plausibility that it will exceed a Required Intercept Rate

Given those results, a new system design might fare better. In the example, System 1 and System 2 design parameters differ as described in Table 1:

Table 1 Design parameters for two different notional BMD systems

	Number of radars	Number of C2 systems	Number of launchers	Number of interceptors per launcher	Probability of intercept	Flight time of interceptor
System 1	2	1	3	4	0.5	6.5
System 2	3	1	4	4	0.62	5.7

Intuition says that System 2 should be more capable because it includes more radars, interceptor launchers, and better quality interceptors with better intercept probabilities and flight times. Indeed, the belief and plausibility for System 2's RIR shown in Figure 5.

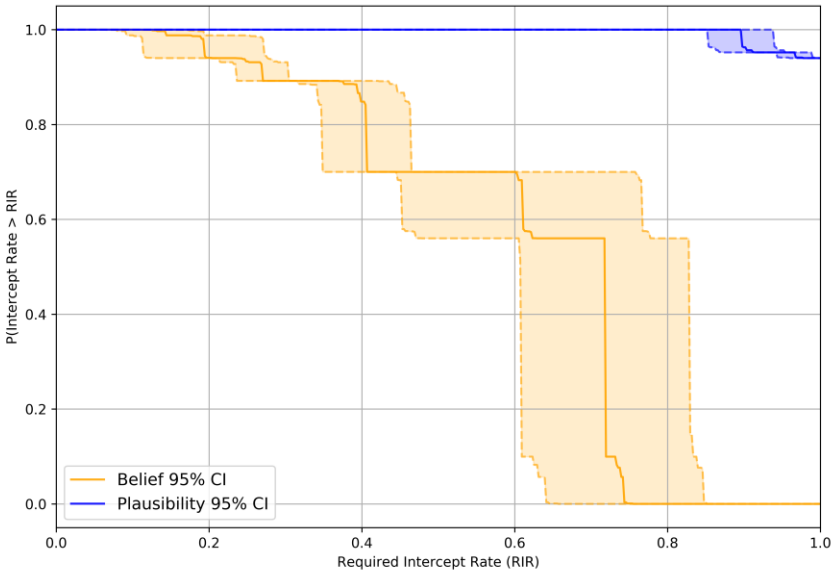


Figure 5 System 2 response variable 'intercept rate' evaluated with respect to belief and plausibility that it will exceed a Required Intercept Rate reflects the overall better performance compared to System 1

Both the Belief and Plausibility values are shifted to the right indicating System 2 as being more capable of meeting the stakeholder RIR requirement.

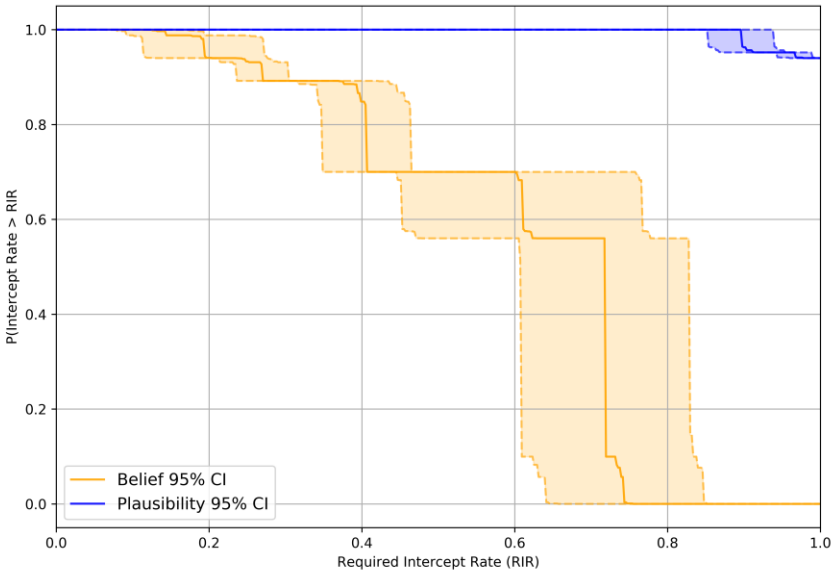


Figure 6 System 2 response variable 'intercept rate' evaluated with respect to belief and plausibility that it will exceed a Required Intercept Rate

Using the same belief BPAs from the experts and combining as before for a BMD system with new design parameters, the plot in Figure 7 is obtained. The results show that if an RIR of 0.82 were prescribed by the key stakeholder(s), only System 2 would have a chance of meeting that performance requirement. This work shows that DST can be used in tandem with non-deterministic system modeling to account for uncertainty across both system performance and operational scenarios that might be encountered as a tool to explore and refine system design parameters.

Similarly, it is not always possible to infer a frequency estimate from prior events. This is especially true for defense and weapons system evaluation problems where past events are based on conditions dissimilar from current and future analytical contexts.

In the absence of information, for example, many methods make assumptions about the nature of the input data, specifically reflecting these assumptions in how the variables are modeled.

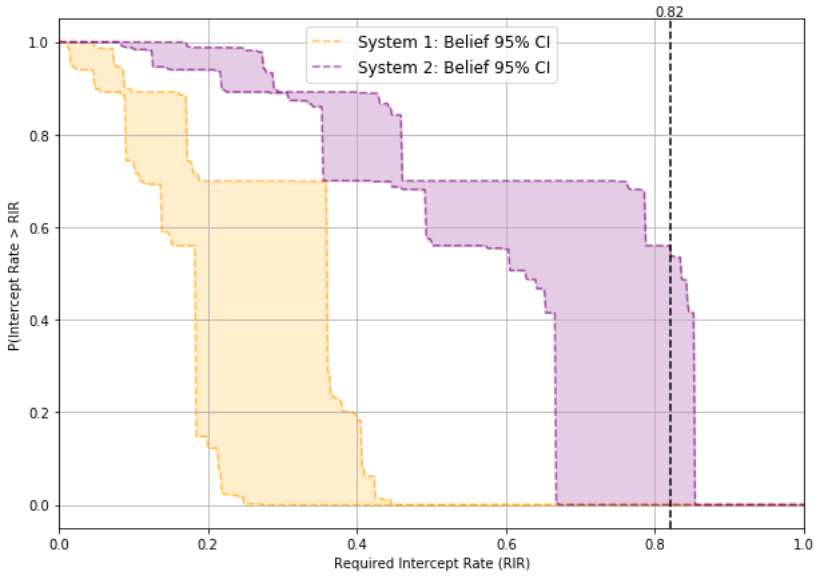


Figure 7 System response variable 'intercept rate' evaluated with respect to belief that it will exceed a Required Intercept Rate for two different system designs

Summary and Insights

This effort constructed a notional model of a simple ballistic missile defense system with control decisions and physical consequences. This simple cyber-physical system model accounts for uncertainty in inputs, which are defined as uncertainty associated with expert opinion regarding likely operational scenarios, and uncertainty associated with system technological performance which was modeled via a stochastic simulation. The intent was to use this model to help evaluate different decision analysis methods that might aid the design process when faced with uncertainty without detailed distribution information. Specifically, the work evaluated methods from evidential reasoning (built from Dempster-Shafer) in contrast to traditional additive value approaches.

The effort successfully used DST in tandem with non-deterministic system modeling to account for uncertainty across both system performance and operational scenarios that might be encountered as a tool to explore and refine system design parameters. In general, the combinations of belief intervals from expert solicitation is a very powerful technique for enumerating the simulation inputs for these stochastic model evaluations as it provides a framework for efficiently sampling the design space and aggregating the results.

Future refinement is needed along two lines associated with this approach. Firstly, there is the issue resulting from the normalization used in Dempster's rule of combination when sources are in significant conflict or when propositions have null values as discussed by [Zadeh, 1986]. Secondly, final belief (B%) and plausibility (P%) values associated with a given outcome may be loosely translated as "Certainly B% and possibly P%". This interpretation, while insightful, may not produce a computational outcome in terms of concretely selecting one design alternative over another.

Regarding the first challenge, [Xu, 2012] discusses an Evidential Reasoning methodology derived from DST that employs a new evidence combination rule established by revising Dempster's combination rule to handle conflicting evidence. The method ascribes a set of weights to the evidence in the combination step, enforcing orthogonality so that the weights sum to 1. Weights are assigned to each of the basic criteria associated with a multi-criteria decision making (MCDM) problem, where each weight reflects the criteria's relative importance to the general criterion. The method seems well-suited to MCDM-type

problems, but further work is required to understand if the approach makes sense to account for conflicts in the type of uncertainty associated with defining potential operational scenarios as was done here. With respect to the second challenge, some of the utility-based ranking methods employed by Xu might add clarity to specifying final decisions. Again, further work is needed to determine if these approaches are synergistic with the types of systems engineering challenges described in this effort, namely those associated with operational scenario uncertainty and non-deterministic system performance.
System models for future defense systems that will be increasingly cyber-physical in nature will require systems engineers to account for multiple critical problem dimensions:

The specific system or set of systems we need to evaluate and its external context so that we can address systemic effects
Uncertainty across relationships, attributes, and drivers.
Changing systems and changing external contexts
Interactions between physical and informational processes

Typically, decision making is viewed strictly as an analytical problem. The process defines alternatives, evaluates them according to predefined criteria and requirements or objectives, and compares the alternatives to find the best performers. This process is well-structured and executable in a computational environment. It can, however, miss insights that different methods can reveal, especially when dealing with uncertainty. Tradespace analyses are needed to support key decision makers and are dominantly based on multi-additive value decision criteria or a variant of this approach. While well-understood and embraced, some aspects of the problem critical to inform design and stakeholder decisions are not well-addressed via these traditional analyses where uncertainty is usually propagated through as simple distributions on input variables. This work sought to evaluate a more subjectivist way of looking at the problem that could be captured and executed in MPTs to advance the future systems analysis needs of the DoD.

ACTIVITY 2: NEXT GENERATION COST MODELING

Overview

This task continued to pilot the development, refinement, and application of SERC cost estimation model ensembles and tools to DoD-system affordability issues impacting system engineering costs in acquisitions programs. GTRI and collaborators from the University of Southern California and the Naval Postgraduate School outlined a SysML cost modeling roadmap in the 2015-2016 report. For Phase 6, the steps for 2017-2018 along that roadmap included maturing this work along the lines of capability extensions using new tools, infusing context into those capabilities, and demonstrating the capabilities in case studies relevant to the DoD.
For the extension work, GTRI proposed to investigate the next round of front-ends based on OpenMBEE and add new functionality to the SysML cost modeling building block library in tandem with the collaborators from University of Southern California and the Naval Postgraduate School. Specifically, the work focuses on creating new building blocks to support (a) software cost modeling (via the COCOMO technique), (b) hardware cost modeling (via the Advanced Missions Cost Model (AMCM) technique), and (c) adding COSYSMO 2.0/3.0 aspects including considerations for reuse, risk, etc. The intent is to address the broader affordability tradespace context and the feasibility of auto-generating inputs to COSYSMO, etc. from a regular SysML-based system model.

Open-MBEE

OpenMBEE [https://github.com/open-mbee] is a SysML-based modeling and simulation environment that is being used in production on several major NASA projects including the Europa mission and the Mars 2020 mission. As one example regarding OpenMBEE scalability and support for major programs, note that the Europa project consists of several hundred people and two main locations (one at JPL in California and the other at APL in Maryland), and that the Europa SysML model consists of 1M+ elements.
OpenMBEE has been developed and open-sourced by NASA JPL, and may be a flexible "model-based wiki" frontend for the cost modeling building blocks. It builds on top of an enterprise-grade content management system (CMS) backend, which supports a common Representational State Transfer (REST)-based interface to the SysML-based repository. One key application of this REST-based interface is on-demand/automated synchronization between a SysML model in MagicDraw and the equivalent same SysML model in the backend repository.
Another key OpenMBEE feature is what can be called its "model-based wiki" capability. Its web browser-based wiki-like front end provides the best of both worlds as follows:

It provides an easy, familiar web browser interface to all members of the project team (such that knowing SysML is not required for a project member to contribute to the project). The web page views are presented in a familiar document-like structure including sections, tables, text, graphics, etc. All aspects are generated from the SysML, and many of those aspects are editable by end users such that they synch back into the SysML model (assuming the user has proper read/write permissions).

It leverages SysML to formulate the single consistent model backend, which represents the rich structured content that is inherent to complex engineered systems.

In Phase 5, the GTRI-USC-NPS team investigated implementing an initial OpenMBEE "model-based wiki" frontend for the cost modeling building blocks. The envisioned ultimate result is an easy-to-use web browser front-end that provides the rich SysML-based backend capabilities to a general audience (i.e., they can take advantage of SysML without having to learn SysML).
Another benefit is that the same technology enables model-based document generation (such as PDFs). Therefore, the team is also implementing an initial version of templates to create typical cost modeling summaries as PDF documents for multiple purposes (distribution to people not familiar with SysML, long-term archiving, usage in traditional document-based processes, etc.). That way you can take advantage of the rich modeling in SysML as well as the generation of familiar and ubiquitous pdf-based artifacts.

Phase 6 Next Generation Cost Modeling Activities

GTRI's activities for Phase 6 will be reported to and included in the technical sections of the Next Generation Cost Modeling activities together with the contributions from the collaborators at the University of Southern California and the Naval Postgraduate School. This will promote better continuity and avoid duplication for the final report.

Next Steps – SQOTA Phase 7 Proposed Activities

GTRI proposed to extend GTRI's SQOTA activities for a final phase, Phase 7, to investigate modeling methods that may be used in analysis tools that address current and newer SERC priorities established by the DoD.

The SERC's 2019-2024 Technical Plan strongly emphasizes improving Security, Safety, and Interoperability of systems participating in multiple, increasingly rapidly evolving systems of systems. Consequently, SQOTA team members have been asked to incorporate concepts of Security, Safety, and Interoperability to the research focus in Phase 7 as they relate to the acquisition process and design phases with associated decision analysis included in that process. GTRI will support that direction by investigating modeling methods most suitable to capture functional capabilities of cyber-physical systems, specifically viewing security as an application that can in turn produce abstract views of dynamic processes of those systems in a computational environment.

SQ Methods, Processes, and Tools (MPTs) Piloting and Refinement

This task will continue to pilot the development, refinement, and application of SERC methods, processes, and tools to DoD-system SQ tradespace and affordability issues. For Phase 7, GTRI will investigate two critical aspects of analyzing capabilities of cyber-physical systems within a computational environment as part of a larger toolset:

Investigate structural concepts of cyber-physical system models most amenable to dynamic simulation. Prior SERC investigations have tended to converge on the notion of representing cyber-physical systems as graphs or directed graphs. These include the System of Systems and Analytic Workbench effort led by Purdue, the System Aware Security work led by the University of Virginia, and a recent pilot investigation lead by Georgia Tech. To conclude GTRI's MPT focus under SQOTA, the Phase 7 work will use these prior investigations as a foundation, evaluating the methods and constructs used, and also consider different approaches in literature to determine the best type of graph structures that will enable dynamic simulation. These structures may be a hybrid between Bayesian and Markovian formalisms or a realized application of Probabilistic Relational Model formalisms, the latter of which is effectively a relational graph. Specifically, the work will investigate which graph structures are best suited to simulate dynamic processes occurring on the graphs. These dynamic processes can be used as an abstraction to help evaluate the effectiveness of a given graph structure with different node attributes at producing or preserving a given capability.

Simulation approaches most suited for increasing problem scale. Once a promising modelling approach is identified, simple graphs will be created and, from these, the work will conclude with an investigation of simulation methods with respect to their ability to efficiently capture the dynamic processes and scale to larger structures.

Together, this work will complete GTRI's work under SQOTA by developing an understanding of the next evolution of these MPTs for the SERC's focus in Security, Safety, and Interoperability.

ILITIES SEMANTIC BASIS

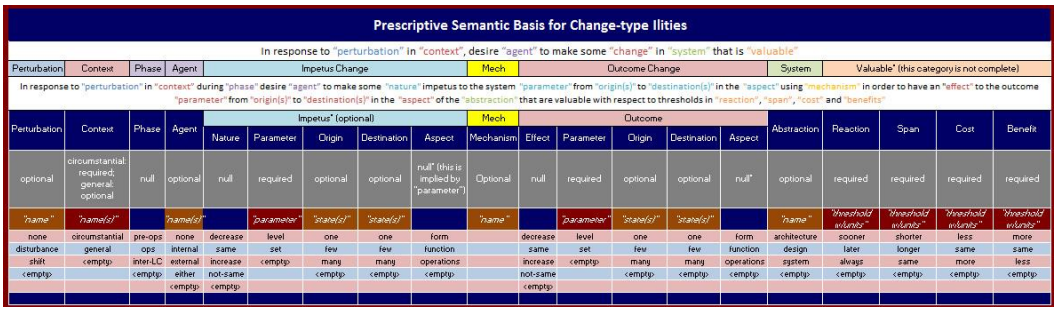
MIT has proposed the use of a semantic basis that can serve as a framework for formulating ility "definitions." Such a basis would provide a common language that would inherently demonstrate how various ilities relate to one another (Ross, Beesemyer & Rhodes, 2011) and provide an opportunity for discovering new ilities as well as provide a new representation for meaning of various ilities beyond English definitions. The Ilities Semantic Basis has evolved over the past several years, based on prior MIT research that aimed to develop a radical new approach for defining ilities (or, SQs, as used in this report), rather than simply proposing yet another set of definitions (Ross, et al., 2011). Prior SERC reports provide detailed discussion of the work to date. A brief summary is provided below to give the reader background information to understand the phase 6 accomplishments. Further description of the semantic basis can be found in the SERC technical reports for the prior five phases.

PRIOR PHASE RESEARCH BACKGROUND

One of the fundamental challenges for developing a clearer understanding of the semantics of "ilities" is the current ambiguity in these terms, which are often used colloquially and therefore inherit informal meaning. Use of "ilities" in some technical disciplines extend from well-accepted prescription; however this has not yet occurred in the systems community as evidenced by the abundance of definition offering papers with conflicting meanings. Additionally, these terms display both polysemy (having multiple meanings that are semantically related) and synonymy (multiple terms having similar meaning). An example of polysemy is two different, but related meanings for flexibility: "able to be changed" and "able to satisfy multiple needs. An example of synonymy is the interchangeable use of terms such as flexibility (able to be changed) and changeability (able to be changed or change itself). One root cause of the ambiguity in technical usage of ilities is that typically ilities are considered one at a time in the literature. Our MIT research and other recent works suggest considering sets of ilities has merit (Ross, Rhodes & Hastings, 2008; Ross, Beesemyer & Rhodes, 2011; de Weck, Ross & Rhodes, 2012; Ross & Rhodes, 2015).

Our research indicates that at least three semantic fields may exist for the general set of system "ilities" including change-type ("flexibility", "agility"), architecture-type ("modularity", "interoperability") and new ability-type (e.g., "auditability"). Identifying and classifying the in- use ility terms into appropriate semantic fields serves to eliminate ambiguity in meaning, usage, and application, as well as allow for the explicit consideration of trade-offs within the semantic field. A consistent basis within a field can allow for direct comparison of its members. Revisiting the concept of relationships amongst the ilities, the basis can provide a first order approximation to clarify semantic differences amongst ilities within a particular semantic field. For example, we differentiate "flexibility" and "adaptability" in terms of whether the change agent is external or internal to the system's boundary, respectively. The basis can also show how a given change statement can display multiple ilities simultaneously. For example, agility

relates to how quickly a change can be executed, so one could desire an agile, scalable change to describe a quick and level-increasing system parameter change. A working hypothesis in our research is that that "architecture-type" ilities are enablers for "change-type" ilities (Lockett, Swan & Unai, 2017).
A prior ilities study looked at co-occurrence of ilities terms in the literature, with implied dependence amongst terms, resulting in a directed graph that tempts reading causal relationships into the links (de Weck, Ross & Rhodes, 2012), but the existence of "co-occurrence" cannot describe the nature of the link. As a complementary approach to discovering relationships amongst ilities, prescriptive assertions can be based upon theory or experience, making conceptual leaps in proposing how ilities should relate to one another. Building upon the insights from looking at various approaches for describing ilities, and drawing inspiration from the linear algebra concept of a basis as a spanning set that defines a space, prior research proposed a prescriptive semantic basis for consistently representing changeability-type ilities within a particular semantic field. Initially conceived as a ten-category basis, the current semantic basis is comprised of twenty categories Figure 8, is believed to span change-type ility semantic field.



Prescriptive Semantic Basis for Change-type Ilities

In response to "perturbation" in "context", desire "agent" to make some "change" in "system" that is "valuable"

Perturbation	Context	Phase	Agent	Impetus Change	Mech	Outcome Change	System	Valuable* (this category is not complete)

In response to "perturbation" in "context" during "phase" desire "agent" to make some "nature" impetus to the system "parameter" from "origin(s)" to "destination(s)" in the "aspect" using "mechanism" in order to have an "effect" to the outcome "parameter" from "origin(s)" to "destination(s)" in the "aspect" of the "abstraction" that are valuable with respect to thresholds in "reaction", "span", "cost" and "benefits"

Perturbation	Context	Phase	Agent	Impetus* (optional)					Mech	Outcome					Abstraction	Reaction	Span	Cost	Benefit
				Nature	Parameter	Origin	Destination	Aspect	Mechanism	Effect	Parameter	Origin	Destination	Aspect					
optional	circumstantial: required; general: optional	null	optional	null	required	optional	optional	null* (this is implied by "parameter")	Optional	null	required	optional	optional	null*	optional	required	required	required	required
"name"	*"name(s)"*		*"name(s)"*		*"parameter"*	*"state(s)"*	*"state(s)"*		*"name"*		*"parameter"*	*"state(s)"*	*"state(s)"*		*"name"*	*"threshold value(s)"*	*"threshold value(s)"*	*"threshold value(s)"*	*"threshold value(s)"*
none	circumstantial	pre-ops	none	decrease	level	one	one	form		decrease	level	one	one	form	architecture	sooner	shorter	less	more
disturbance	general	ops	internal	same	set	few	few	function		same	set	few	few	function	design	later	longer	same	same
shift	<empty>	inter-LC	external	increase	<empty>	many	many	operations		increase	<empty>	many	many	operations	system	always	same	more	less
<empty>		<empty>	either	not-same		<empty>	<empty>	<empty>		not-same		<empty>	<empty>	<empty>	<empty>	<empty>	<empty>	<empty>	<empty>
			<empty>	<empty>						<empty>									

Figure 8 Welcome screen with ilities dictionary choice

While a subset of the basis can be used to generate simpler statements, using the twenty categories provides a richer statement. Applied to the example of changeability of a stereo system, the resulting statement is shown below in blue text. Note: The grey italicized text is included here to show the mapping to categories from the basis.

In response to a loud noise (*perturbation*) late at night (*context*), during operations (*phase*) of system, desire owner (*agent*) to be able to ***impetus*** {increase (*nature*) the knob angle level (*parameter*) from one state (*origin*) to many states (*destination*) in the system form (*aspect*)} through turning the knob (*mechanism*) that results in the ***outcome***{increasing (*effect*) the volume level (*parameter*) from one state (*origin*) to many states (*destination*) in the system function (*aspect*)}in the owner's stereo system (*abstraction*) that takes less than 1 second (*valuable*).

Figure 9 Constructed ilities statement

The prior phase of research developed a detailed design for a software-based implementation of the evolved semantic basis. This was further developed during this phase of the research,
with a goal of leading to potential deployment to government organizations. The proof of concept translation layer is described in the prior phase report, and is referred to as the "ilities Semantic Translation Layer Assistant" (and is NOT operational software).

PHASE 6 ACCOMPLISHMENTS

During Phase 6, the research team further tested the translation layer assistant concept. In this concept, a user decides to use the translation layer assistant; the first step is to pick an ilities dictionary to use. Presently, there are a number of different definition sets for ilities, and the prototype allows the user to make a choice consistent with their own organization. The prior phase report presents conceptual screen captures of the translation layer assistant. Figure 10 is the welcome screen of the assistant.

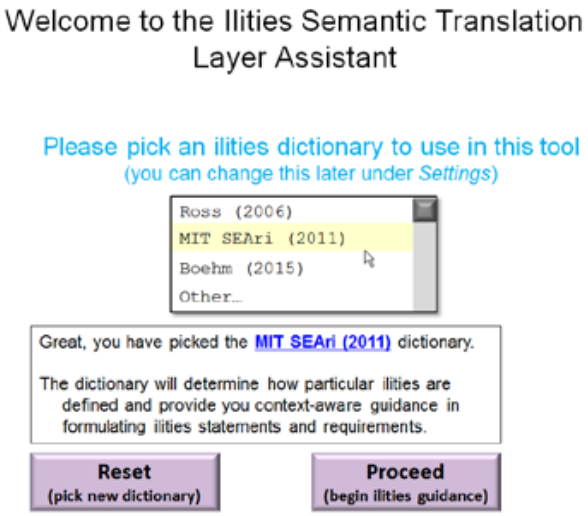


Figure 10 Welcome screen with ilities dictionary choice

The various elements used in the Ilities Semantic Translation Layer Assistant prototype are in Figure 11. Underlying the prototype are core supporting constructs, shown in Figure 12.

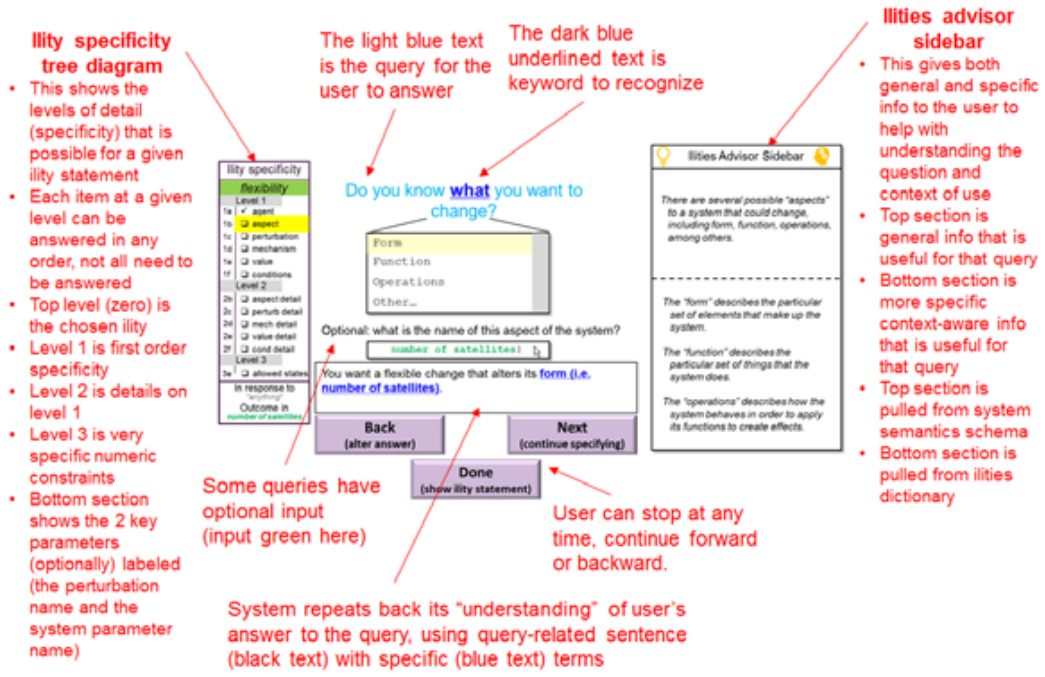


Figure 11 Elements in the Ilities Semantic Translation Layer Assistant prototype

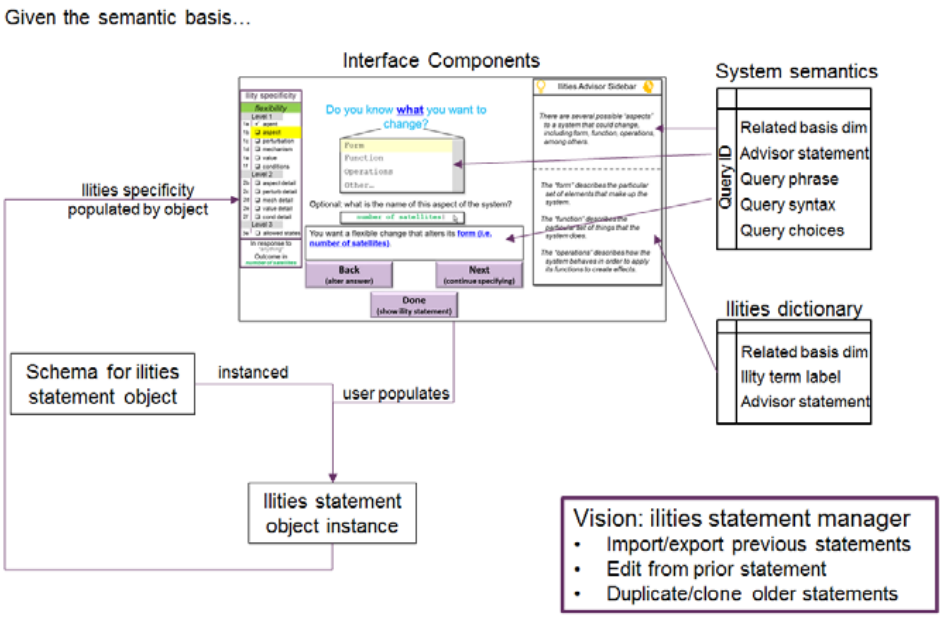


Figure 12 Core Supporting Constructs for the Ilities Semantic Translation Layer Assistant

The conceptual approach for the translation layer assistant was further tested through several sessions with MIT researchers and graduate students. This investigation confirmed potential usefulness of such a translation layer, and small adjustments were made based on the informal sessions. The scope of the

research project precluded more formal testing with a broader stakeholder community. A next step would be to perform such testing prior to developing and building a software prototype.

METRICS DERIVED FROM THE SEMANTIC BASIS

One of the beneficial outcomes of using the ilities semantic basis as a common representation of ilities is that it can indicate a set of distinct, but related, metrics for measuring three aspects of a given ility: whether it is present, the degree to which it is displayed, and the value of that ility. Since the ility term labels map to particular (potentially overlapping) basis choices, the metrics most cleanly map to the basis choices, and then can be post facto assigned to ility terms. For example, scalability (ability to change the level of a system aspect from one state to another) can be measured in terms of number of states accessible, either as a set, or on a per parameter basis. In fact, the basis can be partitioned into three types of defined factors that impact potential metrics: *1) antecedent descriptions (5 categories), 2) state counting (11 categories), and 3) path valuation (4 categories).* See Figure 6 for how these three map to the 20 basis categories.

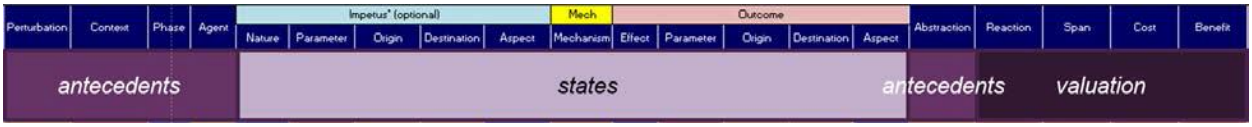
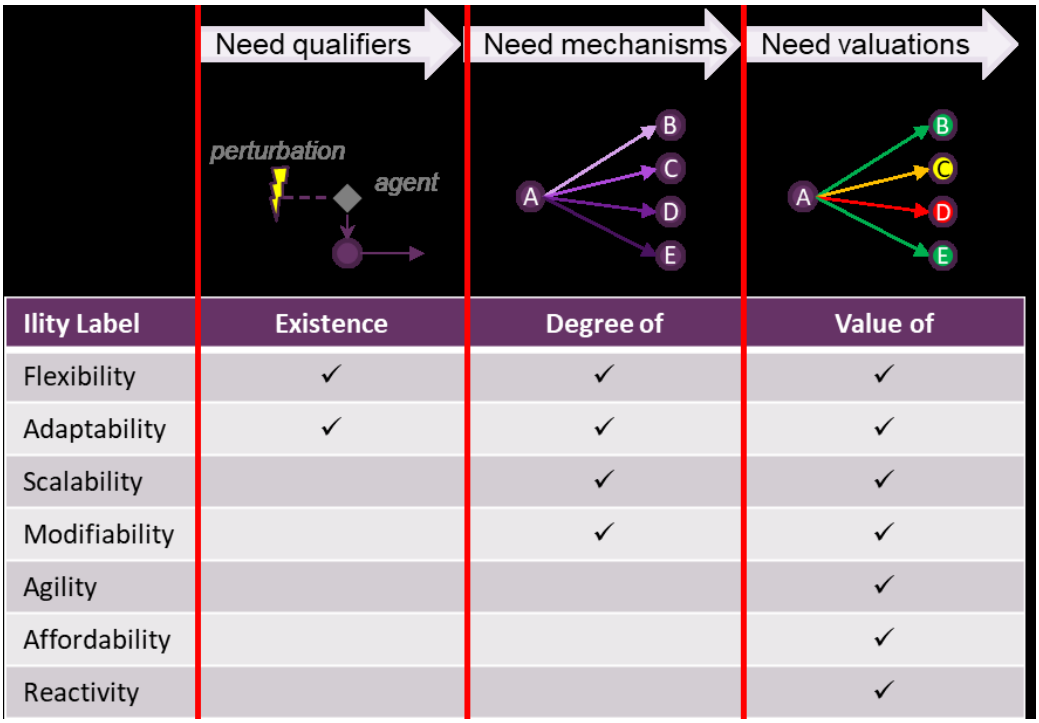


Figure 13 Semantic basis as supplying information for antecedent description, state counting, and path valuation

Using this perspective, we can derive appropriate metrics (existence, degree, or value) for a given ility described by a particular ilities statement using the basis. Figure 14, for example, illustrates several ility labels and whether their specification is sufficient for existence, degree, or value metrics. In this example, flexibility and adaptability relate to antecedent categories (i.e. agent) and therefore are a description of the existence. Their presence in an ility statement is sufficient for existence, but not sufficient to describe degree. For that, we also need to be able to count states (e.g. if we had change mechanisms that describe reachable states). If state counting is supplied, then scalability can be measured. In this example, agility, affordability, and reactivity are defined relative to acceptance thresholds and therefore additional information is needed to measure these (i.e. valuations, how much execution time, cost, and activation time is required for a given change).



Ility Label	Existence	Degree of	Value of
Flexibility	✓	✓	✓
Adaptability	✓	✓	✓
Scalability		✓	✓
Modifiability		✓	✓
Agility			✓
Affordability			✓
Reactivity			✓

Figure 14 Example ility label relation to existence, degree, and value of state changes

As can be seen in the states group of categories, a fair number of basis dimensions relate to state descriptions (e.g. starting point, ending point, desired ranges, etc). Figure 15 illustrates this state (point/range) from/to relationship that can be described in the basis, and will fundamentally relate to assessing the "degree of" various ilities.

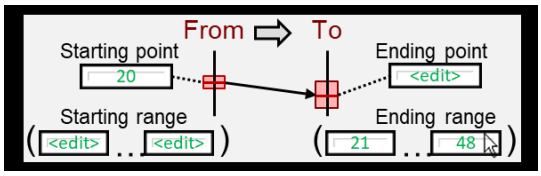


Figure 15 Example ility label relation to existence, degree, and value of state changes

As a practical workflow, consider the following: First the system aspect is defined (the state variable) and then (if a tradespace network is available via described change mechanisms) the number of reachable states where that aspect changes can be counted. A series of optional filters can be layered on top of this representation in order for the metric to capture the intent of that ility. For example, if scalability in number of satellites (system aspect, state variable) has constraints in terms of value (i.e. time to make change, cost to make change, time to set change in motion, benefit of end state), then the metric would only count end states and paths which do not violate the constraint.
Figure 16 illustrates the concept of selective filtering. In this example, consider a starting state A (e.g. satellite constellation), with the existence of an external agent-actuated change mechanism that allows it to change to four other states (B, C, D, E). The antecedent description specifies the existence of flexibility. The existence of states B, C, D, and E allow for "degree of" assessments (here the outdegree is 4). If we desire scalability, then we look at the end states B, C, D, E and only count those that display a change (up or down) in a state variable of interest (e.g. number of satellites in our constellation). Let's

say state A has 10 satellites, B has 10, C has 15, D has 20, E has 25, then we do not include state B in the filtered outdegree assessment. Our flexibly scalable score would then be FOD=3. But let us suppose we want an affordably flexible scalable change and our change cost threshold is "must cost less than $18M" for change. If path A-C is $10M and A-D is $15M and A-E is $20M, then we do not count path A-E, so this results in FOD=2. So, in summary, this simple tradespace network has described a flexible constellation whose degree is 4, but whose flexibly scalable degree is 3 and whose affordably flexible scalable degree is 2. In this way we can trace ilities to metrics and compare alternatives on a common basis.

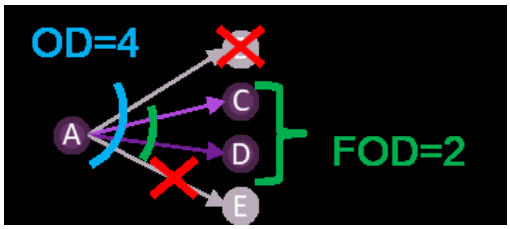


Figure 16 Example selective filtering of outdegree

Since the change-related ilities fundamentally relate to change (i.e. over time), they are particularly useful concepts within the Epoch-Era Analysis (EEA) framework. In particular, multi-epoch analysis lends itself to assessing the impact of changing perturbations on states and their values. This means robustness can be assessed across considered epoch shifts (i.e. perturbations), and the value of changeability can be assessed as well (since we may have additional context-dependent information on the value of end states). The valuation approach to strategic changeability (VASC) is one example method for leveraging EEA to develop quantitative measures for the ilities.

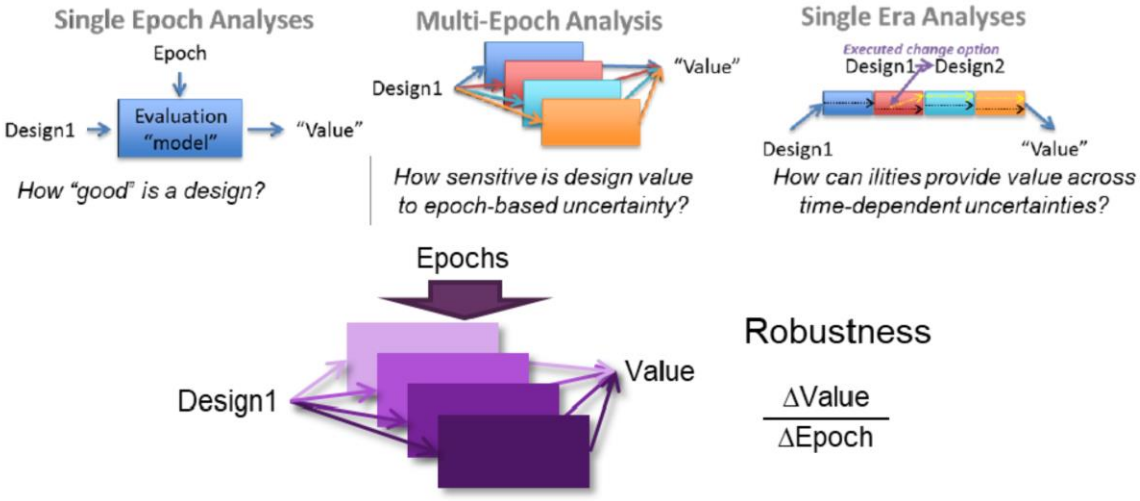


Figure 17 Example use of multi-epoch analysis to quantify robustness

Figure 17 illustrates how analyses can be layered in order to provide sufficient data to quantify variation in metrics in response to epoch and era changes. In particular, perturbations and contexts dimensions within the semantic basis map to epoch shifts and era progression in EEA. State variables of interest (e.g.

form or performance) can be used along with measures of value to determine the degree and value of being insensitive to considered uncertainties. In this way, robustness is a counter point to changeability (i.e. "don't change") yet at the same time enabled by changeability (i.e. execute change options in order to preserve outcome state variables within some tolerance). The additional nuances of path valuations (i.e. how long does it take to execute a change) can be highly context dependent (consider across epoch variations) and path dependent (consider across unfolding eras), therefore using an EEA framework can lend credibility and fidelity to assessments of ilities.

POTENTIAL OPPORTUNITIES FOR APPLYING THE SEMANTIC BASIS

During this phase of the research, the team explored two potential opportunities for applying the semantic basis. The first opportunity is using the existing basis for educating students and practitioners. The second opportunity would involve further research and method development for the purpose of automated technical document comprehension of system qualities ("Ilities") using a prescriptive semantics basis approach.

Education on the Ilities

Formal education of systems engineers takes place at many universities across the globe. Additionally, there is a vast body of continuing training available to practicing systems engineers. Given the lack of accepted common definition of ilities, or system qualities, engineers learn the definition that is used by the educating organization or instructor. Naturally, this is a negative influence of having harmonized terminology. Additionally, the student learns the definition as written, and often does not acquire a more organic understanding of the term.

In this phase of the research we explored the possible application of the semantic basis approach to education. The semantic basis appears to be useful for purposes of discussing ilities in a more general manner. In the classroom, the semantic basis could be used to construct ilities statements using text from a document, to develop a statement such as illustrated in Figure 8. Our research suggests the semantic basis could be a useful approach to developing a more precise understanding and appreciation of what is required in an ilities statement. Use of various versions of the basis (e.g., 10-dimension basis, 14-dimension basis) and classroom use of a translation layer assistant could provide a means to show how simple statements can be logically elaborated as more information is gained.

Automated Technical Document Comprehension of System Qualities ("Ilities")

Defense systems face unprecedented uncertainties, emergent threats, and increasing need for interconnectivity, which necessitate rapid discovery and delivery of responsive capabilities through new technologies and repurposing of existing capabilities. Desired and emergent system properties ("ilities" or "system qualities") are universal across domains, enabling system responsiveness on multiple timescales. While ilities sometimes emerge unexpectedly in operational systems, the ability to design ilities properties into our systems in advance will be the true game-changer. The lack of a common basis for ilities across domains and disciplines, however, inhibits our ability to predictively make decisions to impact these properties. This limits the discovery of the larger patterns that enable individual and cross-domain design decisions that result in the desired ilities manifesting at the systems level. While ility terms are increasing prevalent in technical documentation for defense systems, their precise description remains elusive. Flexibility, agility, and resilience are three examples of the numerous ilities that are often imprecisely articulated and comprehended differently across stakeholder communities, disciplines and

domains. The nature of modern defense systems necessitates being able to translate ilities information across an increasing number of domains and disciplines (e.g. computer science, biotechnology, politic science, cognitive science), driving the ultimate need for a 'Rosetta Stone".

Given this motivation, the prescriptive semantic basis appears to be a useful structure for supporting automated extraction of a comprehensible ilities statement from technical documents. Further, research in the computer science community opens new possibilities for automatic extraction and methods to structure and analyze a comprehensible ilities statement (e.g., Natural Language Processing (NLP) for extraction; Lightweight Formal Methods (LFM) for structuring and analyzing).

Such a proposed application of the semantic basis has the potential to impact DoD capabilities in several ways. First, the automated extraction of comprehensible ilities statements during capability requirements assessment can reveal and enable comparison of ilities (e.g., agility,
modularity) found in documentation for existing and planned systems, thereby informing acquisition decisions. Second, the ability to rapidly generate design concepts in support of repurposing capabilities could be facilitated by generating options through automated composability of ilities semantic fields (akin to a recommender system for the systems designer/decision maker). Third, there would be potential to augment human decision making with curated libraries of ilities statements that can be queried with questions normally taking significant time to investigate. For example: What are all the facets of how my current system supports agile operations? What architecture ilities (e.g., modularity) have been used in space communications systems to enable resilience?

Application of the semantic basis for this purpose may enable the following: (1) ability to automatically extract technical content from text documents using an ilities prescriptive semantic basis, yielding rich, non-ambiguous ilities statements that convey precisely what is meant; (2) ability to automatically synthesize change-ilities and architecture-ilities semantic field information from multiple documents to generate new design concepts (e.g., concepts for achieving a specific aspect of system flexibility through modularity approaches). A potential strategy for achieving this would need to involve multiple disciplines to contribute to extending the existing knowledge and collaborate on a method. This is anticipated to involve the following activities:

1. **Extend fundamental knowledge of ilities to refine and extend ilities semantic fields.** This activity would aim to gain a broader understanding of how ility terms are used and comprehended by investigating cross-discipline/domain sources and elicited expert knowledge from different stakeholders. Results would be then used to refine ilities semantic fields.

2. **Investigate automatic extraction and analysis of ilities statements.** This activity would involve the investigation of useful methods (e.g., NLP, machine learning, LFM) to automate the extraction of ilities statements from various types of technical documents.

3. **Demonstrate automated generation of comprehensible ilities statements.** This activity would evaluate effectiveness (using theory-informed measures) of resulting prescriptive semantic basis approach via demonstration case involving diverse technical documents.

4. **Investigate approach for generating new design concepts through automated synthesis.** This activity would investigate how to automate composability of architecture/change-type ilities semantic fields to generate new design concepts for achieving a desired system property (e.g. new design concepts for system agility through myriad modularity strategies).

Usability and benefits of this proposed method could then be evaluated through application case to evaluate usability regarding automated generation of comprehensible ilities statements and impact to system, and to evaluate stakeholder benefit of ability to automatically generate new design concepts. Further research could employ a designed experiment to understand essential human touchpoints and

other factors that can enhance comprehensibility during automated ilities statement generation and subsequent human decision making. Additionally, it is envisioned that the work could draw on research on extracting information from non-text
modalities (pictures, graphs), in order to investigate an approach for using non-text-based information to augment ilities semantic fields.

EXTENDING THE BASIS

During Phase 6, the research team continued discussion of the semantic basis research in various technical exchanges with the systems community, including SERC, FFRDCs and other research stakeholders. The research team has explored relationships between the semantic basis and the USC work on system qualities. This has included leveraging early work by USC to map their system qualities list to the semantic basis, here focused on six out of the twenty basis categories. Figure 18 shows the initial draft mapping of the USC system qualities (ility labels) to the six basis categories. Of note is the modification of specified choices for the basis categories. Presumably this is driven partly by USC's software development experience (e.g. aspects as "form" or "scope" as opposed to "form", "function", "operations").

Perturbation	Agent	Impetus		Outcome	
		Nature	Aspect	Effect	Aspect
disturbance	internal	increase	form	same	form
defect	external	decrease	scope	not-same	scope
opportunity	<empty>	re-host	<empty>	reduced	<empty>
<empty>		<empty>		increased	
				<empty>	

						USC Ility Label
disturbance		increase		same, increased		Resilience
				not-same		Changeability
disturbnce	internal			same		Robustness
disturb, opp'y	internal			not-same		Adaptability
defect	internal			same		Fault-Tolerance
defect	internal			not-same		Self-repairability
disturb, opp'y	external			not-same		Modifiability
defect	external					Repairability
	external					Maintainability
	internal			reduced		Graceful Degredation
			form	not-same	form	Reconfigurability
disturb, opp'y		increase	scope	not-same	scope	Extensibility
disturb, opp'y		decrease	scope	not-same	scope	Contractability
disturb, opp'y		re-host	form	same	form	Portability
disturb, opp'y			scope	same	scope	Versatility
						Survivability
						Evolvability
						Flexibility
						Agility
						Reactivity
						Exchangeability
						Variations by

Figure 18 USC ility term (initial draft) mapping to basis

Taking these initial mappings, we compare the USC to the MIT semantic categories choices in Figure 19. This is only an initial comparison, and it highlights a few points to consider in the next phase of research. This includes different choices for basis categories, potentially different interpretation of the meaning of "disturbance," a set of system quality terms that are not the same, and potential interpreted differences arising from MIT's hardware-centric perspective and USC's software-centric perspective. Reconciling these perspectives has promise for further maturing the underlying basis and its applicability across a broad range of system types.

Perturbation	Agent	Impetus* (optional)		Outcome		Reaction	Span	Cost	Benefit	MIT ility Label	
		Nature	Aspect	Effect	Aspect					USC ility Label	USC-MIT Differences
	either	not-same		not-same						Changeability	
				not-same						Changeability	Match
shift				same						Robustness	
disturbnce	internal			same						Robustness	Shift vs. Disturbance
	internal	not-same		not-same						Adaptability	
disturb, opp'y	internal			not-same						Adaptability	Match
	external	not-same		not-same						Flexibility	
										Flexibility	External Adaptability?
		not-same		not-same						Modifiability	
disturb, opp'y	external			not-same						Modifiability	Just External (Internal = Adaptability)
		same	form/ops	not-same	function					Functional Versatility	
		same	form/fnct	not-same	operations					Operational Versatility	
disturb, opp'y			scope	same	scope					Versatility	Similar
shift		not-same		not-same						Evolvability	
										Evolvability	Same as Changeability?
	either	not-same		increase						Extensibility	
disturb, opp'y		increase	scope	not-same	scope					Extensibility	Same - Also Scalability-Up
		same	form	not-same	form					Form Reconfigurability	
		same	operations	not-same	operations					Operational Reconfigurability	
			form	not-same	form					Reconfigurability	Similar
disturbance				same						Survivability	
										Survivability	Robustness + Adaptability?
		not-same		not-same		sooner				Reactivity	
										Reactivity	Focus on duration of fix needs
		not-same		not-same			shorter			Agility	
										Agility	Focus on fix speed
shift				same						Value Robustness	MIT only
disturbance				same						Value Survivability	
shift		not-same		same						Active Robustness	
shift		same		same						Passive Robustness	
shift	none	same		same	form					Classical Passive Robustness	
		not-same		not-same						Scalability	
		same	fnct/ops	not-same	form					Substitutability	
disturbance		increase		same, increased						Resilience	Not specified
defect	internal			same						Fault-Tolerance	Not specified
defect	internal			not-same						Self-repairability	Not specified
defect	external									Repairability	Not specified
	external									Maintainability	Not specified
	internal			reduced						Graceful Degredation	Not specified
disturb, opp'y		decrease	scope	not-same	scope					Contractability	Not included - Also Scalability-Down
disturb, opp'y		re-host	form	same	form					Portability	Not specified
										Exchangeability	Like Modifiability?
										Variations by	Value, Form, Function, Operation

USC only

Figure 19 Comparison of USC (initial draft) to MIT ility term label mapping

In the next phase of research, the team will develop a more detailed mapping between the semantic basis and the USC developed means-ends framework.

NPS RT-181 Final Report

Two NPS Phase 6 contributions to the SQOTA project. The first is an NPS Ph.D dissertation analyzing the prospects of more product-line engineering of Navy systems, by LT Rob Hall. The second is a joint Naval Center for Cost Analysis-NPS-USC paper addressing the challenge of cost estimation for DoD software systems using increasingly agile development via early domain-specific modeling. It received the Best Paper award at the Empirical Software Engineering and Measurement Conference in November 2017.

PSU RT-181 FINAL REPORT

Under PSU's DoD-funded project, *Product Architectures, Design, & Manufacturing for Operational Responsiveness*, DASD (R&E), PSU designed, built, and flew 3D printed UA's, completing the project in August 2017. In the effort, we instrumented the design process, gathering metrics on the design evolution to include total man-hours, models used, and design iterations. PSU used the RT-181 Phase 6 effort to collate the data and publish it to the design community for study. During this period, PSU collated and organized the data set, and prepared a conference paper for submission to the ASME IDETC 2018 conference**.**

USC RT-181 FINAL REPORT

USC's Systems Quality Ontology has been useful in addressing several challenging issues involving system qualities. In his keynote address at the 2018 International Conference on System Resilience in February 2018, Dr. Boehm was able to use the Ontology to elaborate on the best available definition of system resilience in the INCOSE Systems Engineering handbook, in identifying the various forms of Resilience and their relationships to Dependability and Changeability in the Ontology. He extended the presentation to a tutorial for the Aerospace Corp.'s 2018 Ground Systems Architectures Workshop in March 2018, whose theme was Systems Resilience. The abstract for the tutorial, "Life Cycle Resilience Depends on Maintainability," is provided below. The Ontology was also presented at the International Workshop on Software QUAlities and their Dependencies (SQUADE 2018) at the 40th International Conference on Software Engineering. In May 2018.

The Ontology's identification of Maintainablity as a critical means to not only Life Cycle Efficiency via Total Ownership Cost (usually about 75% spent on Maintenance), but also Changeability (internal via Adaptability or External via Modifiability or Repairability) and Dependability (via its contributions to Availability from Mean Time to Repair). USC has also made major contributions to the analysis of Maintainability and Technical Debt via large-scale data analytics such as the Software Quality Understanding by Analysis of Abundant Data (SQUAAD) toolset. It has helped the Navy address some system safety concerns, and has found ways to find and reduce Technical Debt during development on large software-intensive systems. SQUAAD's primary developer, Pooyan Behnamghader, presented early results at the 2017 SERC Systems Research Review, larger-scale results at CSER 2018, and has submitted wider results to the 2018 Empirical Software Engineering and Metrics conference in a paper included below.

Further in-depth analyses of Technical Debt have been documented in two papers led by Reem Alfayez on developers' influences and a paper led by Celia Chen on root-cause analyses, cited below. Another short paper by Boehm and Behnamghader on non-technical sources of Technical Debt and a Software Maintenance Readiness Framework for monitoring a system's maintenance readiness during development. is included below.

Reem Alfayez, Pooyan Behnamghader, Kamonphop Srisopha, and Barry Boehm. 2018. An Exploratory Study on the Influence of Developers in Technical Debt. In Proceedings of the 1st International Conference on Technical Debt. 1–10.

Reem Alfayez and Barry Boehm, Who Introduced this Technical Debt, When, and Why? A Fine Grained Approach for Analyzing the Source and Timing of Technical Debt

Celia Chen, Lin Shi, Michael Shoga, Qing Wang, and Barry Boehm, How Do Defects Hurt Qualities? An Empirical Study on Characterizing A Software Maintainability Ontology in Open Source Software, Proceedings, QRS 2018.

Pooyan Behnamghader, Barry Boehm, Kamonphop Srisopha, Patavee Meemeng, Iordanis Fostiropoulos and Di Huang, Towards Enabling a More Complete Byte-Code Analysis Over Commit-History, submitted to ESEM 2018.

USC's Phase 6 contributions to cost estimation include joint results with the Software Engineering Institute (SEI) on causal cost modeling, Barry Boehm's keynote presentation at the March 2018 SEI Software and Cyber Solutions Symposium of a version of the "Early Phase Cost Models for Agile Software Processes in the US DoD" paper attached to the NPS section, and Jim Alstad's Bayesian calibration of the COSYSMO 3.0 systems engineering cost model, to be presented at the August 2018 Navy-NGA-sponsored IT-CAST cost estimation workshop, and at the meeting of the ARDEC-sponsored Practical Software Measurement Users Group in September 2018.

Besides USC's interactions with the Aerospace Corp. and the Software Engineering Institute, the Mitre Corp. is very interested in using USC's maintainability data analytics, System/Software Maintenance Readiness Framework, and the COSYSMO 3.0 cost model, based on Dr. Boehm's presentations and discussions in an April 2018 visit to Mitre-McLean personnel. We did a live distance demonstration of the SQUAAD toolset to Mitre personnel in May; they are now identifying potential user projects, and we have a followup visit to Mitre in July.

U. VIRGINIA RT-181 FINAL REPORT

UVa has continued to develop a stochastic heuristic search technique for optimizing system properties, in this case runtime performance with big-data computing stacks, including Hadoop and Spark, the most widely used platforms for big data computing as case study subjects. We have developed and extensively experimentally evaluated a novel approach to optimizing runtime performance of big data stacks using an Evolutionary Monte-Carlo Markov Chain technique, combined with tactics for radical reduction in cost of search through the validated use of small data proxies for big data computations during search, and by validating experimentally the applicability of results for one kind of job for other similar kinds of jobs, where similarity is measured by summarized data on dynamic profiles at the level of system calls. Our overall technique and the experimental results, which demonstrate very substantial speedups relative to common use of default configurations, are described in the attached paper, which has been submitted to the ACM Conference on the Foundations of Software Engineering. Ongoing work includes extension to multi-attribute situations and to the broader range of problems in systems engineering.

WAYNE STATE RT-181 FINAL REPORT

During SQOTA Phase 6, Dr. Gary Witus has been working on both TARDEC and SERC sponsorship on developing next-generation TARDEC Ground Systems Engineering Methods, Procedures and Tools For Autonomy-Enabled System (AES). Phase 6 has integrated findings and results from previous and related research on ground systems model-based engineering and set-based design to address the challenge of developing autonomous behavior requirements for AES. The focus is on requirements for the "cognitive"

aspect of AES: situation awareness, context, goals, plan & behavior adaptation for the situation, coordination with other team elements, consideration of predicted outcomes in behavior adaptation, simultaneous internal supervision simultaneous with execution, and related systems engineering methods, procedures, and tools (SE MPT).

The cognitive behavior requirements are the basis for perception and automatic control requirements. The cognitive behaviors are the source of success or failure in integration with team operations and military utility.
SE MPT to verify and validate the cognitive behavior specifications is a goal outside the scope of this phase. SE MPT to derive subsystem requirements for sensors, actuators, signal processing and automatic control subsystems and components are outside scope.

The objectives of this phase are to apply, test and assess SE MPT for deriving cognitive behavior requirements with (1) an audit trail from doctrinal and operating procedures, (2) an audit trail to detailed design and implementation, and (3) potential for early V&V analysis. It will assess strengths and limitations, investigate issues in transition to end-users, and document the findings. The focus is on applying and testing the MPT.

The plan of work is to apply the SE MPT in two trial applications: TARDEC's Autonomous Convoy S&T project, and training in SE "best practices" for Level-4 autonomous driving project (separately funded by a commercial partner, Hyundai-Mobis, in the Detroit metropolitan area). The goals are to assess transfer and application issues, value and gaps/needs for engineering development projects and providers.

In parallel, WSU is developing a graduate course in "Systems Engineering for Autonomy-Enabled Vehicles" to be offered Fall 2018, based on content from SERC projects, Government and commercial partner collaborations.
The expected outcomes are (1) documentation of well-defined SE MPT for cognitive behavior requirements, (2) a AES global functional architecture model accepted by users, (3) a technical architecture model for AES cognitive behaviors accepted by users, (4) preliminary assessment of capabilities, gaps and limitations of software platforms to express, test, V&V autonomous agent behaviors, and (5) identification of SE R&D needs vis AES acquisition in the DoD context.
The focus of this research is on SE for behaviors and operational behavior requirements. The goal is to provide repeatable and scalable methods to generate autonomous behavior requirements towards multiple objectives:

- Ensure compatibility team operations by deriving requirements from unit training, doctrine, tactics, techniques and procedures to
- Produce executable behavior requirements, i.e., requirements that can be tested and can be directly transferred into AES implementation
- Enable incremental development and refinement of behavior specifications following a "reliability growth" model
- Be based on a sound functional architecture for AES and technical architecture of the "cognitive" component
- Being mindful of the adage "Be careful what you ask for – you might just get it."

This project Has been addressing an unmet need for SE MPT to develop requirements for "cognitive" capabilities and adaptable behaviors ("autonomy") that will mesh with team operations. In some sense, the big-picture goal is to develop an interface standard between human and artificial cognitive agents for situation awareness, goals, plan templates, and behaviors that (1) will ensure compatibility with military

operations and personnel, and (2) suitable to specify autonomy capabilities and derived requirements for contracting purposes. The research objectives have been to

- Demonstrate an AES functional architecture model that insulates "cognitive" capabilities from the sensor, signal processing, actuators and control level
- Demonstrate an effective cognitive architecture based on the Belief-Desire-Intention model
- Assess the results in cooperation with Government and industry transition partners

This phase has built on prior research products and results:

- The three-layer AES model that isolates "cognition and behavior" logic and models from specific sensor, actuator, signal processing, automatic control
- SE MPT for high-level requirements that ensure AES cognitive behavior requirements are compatible with system and team operations components by specifying high-level requirements in terms of tasks, roles, functions and team coordination in training and doctrine materials
- An AES cognitive architecture model for situation awareness, goals, plan refinement/instantiation, behavior execution, and internal supervision of behavior execution based on the preeminent practical reasoning system model for autonomous agents (alternately known as Belief-Desire-Intention, Ends-Ways-Means, Awareness-Goals-Plans-Behavior)
- A model of Situation Awareness that includes expected outcomes over time in the current situation (implying requirements for prediction), identification and classification of situation conditions and agents, estimation of their attributes, assessment of classification error probabilities, assessment of attribute measurement accuracy
- A situated model of constraints, goals, plans and behaviors in which
    - "Situated" means that the activations and details depend on the situation
    - Practical models incorporate outcome constraints, outcome objectives, and conflicts among ways and means
        - Outcome prediction models providing a "safety check"
        - Set-Based Design model as the basis for action/activity selection and specification to defer down-selection to maintain maximum adaptability for situation development
    - Goals are activated based on mission concept and situation
    - Plans are pre-defined "PERT chart" templates with parameters computed for the situation to achieve desired outcomes
    - Behaviors are the action/activity instructions that result from arbitration between incompatible goals and blending compatible goals
        - Due to uncertainty in perception situation classification and measurement
        - Due to incomplete specification of conditions for activation
- SE MPT to derive requirements on sensors, signal processing and data fusion to identify, classify, and assess situations (including expected future outcomes – imposing requirements for invertible prediction models) from the "cognition and behavior" requirements – start to fall out from this model
- Training and doctrine are well documented
    - Assumes human-like perception/reaction/understanding
    - Assumes that human training and skill qualification testing is sufficient
    - Convoy operational procedures, threat attack scenarios and response operations, defensive driving procedures, pre-crash situations, emergency driving, etc. are very well defined from a driver perspective
    - Need is for SE MPT to translate into situated cognitive and behavioral requirements for AES

A summary of the desired results is as follows:

1. Problem and Opportunity
   - Potential Benefits of AES for US Military
   - State of Practice in Military and Civilian AES
   - AES "Cognitive" Behaviors and Components
   - "Cognitive" Behavior Requirements are Different from Traditional Requirements
   - SE MPT Needs for "Cognitive" Behavior Requirements
2. Technical Background – Prior and Related Research Foundations
   - Three layer AES model isolating the cognition function
   - Technical architecture for AES cognition with situation awareness, goals, plans and behaviors
   - Set-Based Design Model with Deferred Decisions in AES Plan Execution
   - Sources and MPT for AES cognition requirements for coordinated team operations
   - Scalable approach to developing AES cognitive capability requirements
   - Assurance argument model
   - Issues and approaches for V&V of AES cognitive capability requirements
3. Technical Objectives
   - Document, illustrate, and assess SE MPT for AES behavior requirements
   - Document further SE research issues in AES development, verification and validation
4. Plan of Work
   - Apply the SE AES behavior requirements MPT in two trial application contexts
   - Review the execution and results with potential end users and transition partners
   - Document the project, results, findings and recommendations
5. Transition assessment and co-development partners
   - TARDEC Autonomous Convoy Project
   - Level-4 Autonomy Project with a Tier-1 Automotive Supplier (Hyundai-Mobis)
     - WSU Academic Initiatives
   - Other external coordination opportunities
6. Risks, Mitigations and Opportunities
   - Choice of software platform to demonstrate the AES cognitive architecture model
   - SE for Heuristic Machine Learning (HML) in AES
   - Verification & Validation of AES Requirements and Specifications
7. Staffing and Qualifications

Body

1. Problem and Opportunity
   - Potential Benefits of AES for US Military
     - Extend the span of awareness and influence (range, density & dwell time)
     - Enable ability to engage in high-risk operations without putting personnel at risk
     - Reduce training time and skill qualification levels
     - Provide faster-than-human response in emergency reaction situations
     - Relieve operators from mundane tasks to concentrate on other tasks
   - State of Practice in Military and Civilian AES
     - Examples of AES in U.S. use, development, and/or consideration
       - High speed reaction: Phalanx gun, Active Protection Systems, Anti-Lock Braking

        - Reducing operator workload: Adaptive Cruise Control (in early fielding), Lane Following (in early fielding), autopilots
        - Planned and coordinated operations: Drone aircraft and satellites, unattended ground and sea sensors
        - Extended reach: Fire-and-forget munitions and sub-munitions, unattended ground and sea sensors
        - Extended endurance: Automatic target recognition in reconnaissance and surveillance video (in process), drone surveillance, facial recognition in security
    - National Highway Safety Administration (NHTSA) model for autonomy-enabled driving
        - Level-2 covers advanced driver assistance systems (ADAS) for specific functions. Safety V&V uses the historical model of crashes per mile and per hour of use, with some analysis of the conditions producing adverse outcome
        - Level-3 involves handoff back-and-forth and coordination between a human operator and autonomy-enabled subsystems in a mixed initiative model. The emerging commercial automotive industry consensus is that these handoff interactions are a significant source of design and operational safety risk, and that Level-3 autonomy should be avoided, skipping from Level-2 to Level-4
- Level-4 provides for situation-adaptive autonomous driving, within specified Operational Driving Domain of driving tasks and conditions.
- AES "Cognitive" Behaviors and Components
    - Focus is on the "cognitive" component of AES
        - Situation awareness, goals plans and behaviors, execution and self-monitoring
        - Situation awareness requirements imply perception and prediction capability requirements
        - Behavior execution requirements imply actuation and automatic control capabilities
    - AES functional performance
        - Doing the right thing at the right time within the mission plan
        - Employing, adapting and executing the appropriate behaviors for the situation
    - AES Failure Modes vis-a-vis Team Operations
        - Fails to conform to mission planning and coordinated execution models
        - Imposes limitations on the operational concept to achieve the mission
        - Imposes excessive coordination burden on supervisors, adjacent agents
        - Fails to be adaptable, being effective only in narrow functions and situations
    - Properties AES Need to Exhibit to Achieve Benefits in Team Operations
        - Be *robust and reliable*: can be relied upon to perform its tasks and functions over a wide and understood range of operational situations, for roles and tasks that involve selecting, adapting, executing behaviors appropriate to the situation, including explicit or implicit coordination with adjacent elements
        - Be *compatible and interoperable* within other elements and planning/execution processes within the Service Training and Doctrine and unit Tactics, Techniques and Procedures (TTP)
        - Be *useable and useful* to enable team operations: perform important functions contributing to unit operation effectiveness to achieve the potential benefits. Instructions to the AES should not require a change to Training and Doctrine or unit TTP but should accommodate it.
        - Be *adaptable and adjustable* for the situation, mission plan, and dynamics – including coordination within the team and neutral/adversarial actions

- "Cognitive" Behavior Requirements are Different from Traditional Requirements
  - Consider a typical requirements for a truck
    - Cargo capacity, seating capacity, horsepower, fuel tank size, acceleration, braking deceleration, ground clearance, side slope stability, fuel economy, etc.
    - All traditional quantitative measures of functional properties we are familiar with
  - Consider requirements for a truck with Level-4 autonomy
  - Suppose the Operational Driving Domain is simple expressway driving
  - How do we want the vehicle to behave when
    - Another vehicle cuts in front
    - An entry ramp is coming up
    - The vehicle ahead is driving slow
    - Lead vehicle changes lane
    - It has just started to snow
    - The trailing vehicle is following close
  - The trailing vehicle in an adjacent lane pulls ahead between the ego vehicle and the lead vehicle
  - Defensive driving principles and "rules of the road" provide guidelines for humans under the assumption that a competent driver will figure out what to do – AES cannot do this
  - For an autonomous system the behaviors must be specified so that OEMs, regulators, customers, insurance companies, and lawyers have assurance of the safety and effectiveness over the scope of the Operational Driving Domain
  - Comprehensive enumeration of conditions with specific rules is infeasible
  - General rules and driving principles will need to be composed, arbitrated and blended to apply broadly
- SE MPT Needs for "Cognitive" Behavior Requirements
  - Cognitive architecture model and requirements expression model for composable behaviors
  - Methods to specify high-level behavior requirements to ensure operational compatibility with the team
  - MPT to derive executable "design to" behavior specifications from high-level requirements
  - MPT to derive perception, sensing, actuation and automatic control subsystem requirements as needed to implement the behaviors
  - V&V MPT to analyze the requirements for conflicts, gaps, ambiguity, etc.
  - An assurance model for offerors to justify their claims for proposal preparation, technical reviews during engineering development, and product acceptance evaluation

2. Technical Background – Prior and Related Research Foundations
   - Three layer AES model isolating the cognition function
     - Platform sensors, actuators, signal processing and automatic control are low-level subsystems whose technologies and designs are rapidly evolving, are not standardized, and can be specific to the particular platform
     - Intermediate layer produces the percepts needed for situation awareness, and goal/plan/behavior implementation needed by the cognitive layer, thus isolating the cognitive behavior layer should be isolated from limitations of specific cyber-physical components
     - Classification confidence and measurement accuracy are part of the percepts
     - Maneuver tolerances are part of the execution instructions

- Technical architecture for AES cognition with situation awareness, goals, plans and behaviors
  - Situation Awareness
    - All of the information needed to select, instantiate, arbitrate/blend and instruct goals, plans and behaviors
    - Requirements derived from desired outcome behaviors
    - Includes agent and situation state detection and classification (with associated confidence), and the measures properties/attributes (with associated accuracy)
    - May include estimates of future situation state and time/distance to state transition without change in AES behavior, and with AES behavior options
    - Role for simple predictive models in the situation awareness object
  - Goal/plan/behavior cognitive architecture model
    - Alternately known as the Belief-Desire-Intention (BDI) model, and Ends-Ways-Means model
    - Formal grammar goal-oriented behaviors (the situation conditions, sub-goal/ways network model, and resource arbitration) and composition algebra are in process, with targeted completion at the end of Phase 6.
      - Situation awareness inputs include perceptions, predictions, and mission-plans
      - In concept, the situation awareness object include the detection and classification of objects, confidence in the detection and classification, measurement-based estimates of the properties of the objects, and accuracy of the measurements
      - In any situation, some fields are knowable and others are inapplicable or have unknown values. This is the sort of uncertainty that engineering development has to deal with. In the case of AES, it is closely linked to the behavior requirements
      - Situation awareness patterns are expressed as a simplified regular expression
      - "AND" over all constraints on all fields
      - Wildcard, negation, set for classification data types, min and max for scalar data types
      - A goal is associated with a "PERT chart" of subgoals, i.e., the template of a plan, and computational functions operating on the situation awareness parameters to calculate the specification and control parameters of the sub-goals
      - Heuristic Machine Learning (HML) has a tremendous opportunity here
      - Well-suited for HML capabilities that compensate for our mathematical formulation limitations, in a limited scope and function, so that V&V of the methodology, procedures and outcome is possible
      - The lowest level of a goal is to initiate an action or activity, where the goal and the means are the same. The execution process is too fast to be monitored. Commitment to "do it" is required.
        - Let's say an action is the activation of an automatic behavior, with appropriate control parameter values
        - An action means no more oversight
        - An activity can have oversight for (the integral role in a PI or PID controller)
      - A goal has activation level depending on the activation of the its parent goal, and the criticality/sensitivity to the parent
      - Low-level goals can be compatible with multiple parent goals

        - Plans for competing parent goals can be initially compatible with deferred down-select decision
        - Valuation of instructions for immediate actions and control loops in the context of alternative possibilities is a hard question, but the valuation model is even harder
        - I have worked down in these weeds
        - I want to promote the muck to knowledge for practical use
    - What outcomes do we want to achieve and/or what states do we want to be in? Can have multiple competing goals due to
        - Overlapping goal/plan/behavior specifications on different dimensions that need to be composed – arbitration and blending
        - Uncertainty in situation classification and attribute measurement
        - Uncertainty in the future behavior and responsive behavior of other agents – teammates, adversaries, neutrals, higher-level commanders
    - What is the pre-defined "PERT chart" of sub-goals or tasks to get there
        - A standard model A precedes B, A and B can be conducted in parallel
        - Plans for goals are about sub-goals sequencing, timing and coordination
        - At the lowest level, a sub-goal is to accomplish an automatic action or automatic control activity, with the corresponding behavior to initiate the action or activity
    - Multiple nested control loops produce supervision and pre-emption at different space and time scales
- Set-Based Design Model with Deferred Decisions in AES Plan Execution
- Sources and MPT for AES cognition requirements for coordinated team operations
    - Existing Training and Doctrine materials are a golden source of what to do when and how
    - In the commercial world, Defensive Driving and Rules of the Road materials fill a similar role
    - This is the context for allocation of duties and responsibilities to man and machine, and commander/supervisor/partner expectations
    - The SE disconnect is that T&D assume certain competencies of an 18-year-old recruit, a recruit that has passed certain skill qualification tests, that is advanced in his/her career based on demonstrated ability to adapt to the situation while applying doctrinal procedures
    - This is the hard nut to crack for AES. We cannot specify what to do in all situations, or even more importantly, how to understand the situation
    - Nonetheless, this is the starting point for AES perception, cognition, and execution requirements
    - At least for autonomy-enabled convoy and autonomous driving we have determined that appropriate and suitable doctrinal materials exist for how to behave in complex situations and conduct complex operations involving teamwork, neutrals and adversaries (specifically in the context of Convoy Operations, see FM 4-01-45, Tactical Convoy Operations, MCRP 4-11.3F Convoy Operations Handbook, and Enemy Tactics, Techniques and Procedures (TTP) In Attacking Convoys, etc.)
- Scalable approach to developing AES cognitive capability requirements
    - Even though we cannot exhaustively enumerate exactly what to do under what conditions, we are still faced with behavior requirements for what to do when and where

        - We need a scalable approach in which we can add behavior guidelines and requirements, and test their interactions with other guidelines and requirements
        - We need approach for composable requirements, meaning that there is an integration/arbitration/blending mechanism built into the SE process
        - The upshot is that we need a "cognitive executive" that can pursue parallel goals with commensurate plans, until the need for an action requires selecting one over another
        - This is the set-based design model, applied to goals, plans and behaviors
    - Assurance argument model
        - The model for safety assurance is an "outer loop" that predicts outcomes and inhibits or accentuates plans
        - The assurance argument is based on a structure of claims supported by assumptions, evidence and sub-claims, and the calculus to combine them
        - This is an evolving domain with respect to contractual obligation, legal arbitration, and regulatory approval
        - Important value is in DoD contracting with vendors
        - Current models are deficient, lacking means to
            - Combine disparate types of evidence
            - Tradeoff scope and confidence
    - Issues and approaches for V&V of AES cognitive capability requirements
        - We have only started to touch on this topic
3. Technical Objectives
    - Document, illustrate, and assess SE MPT for AES behavior requirements
        - Demonstration Context #1: Autonomy enabled convoy operations (TARDEC S&T project)
        - Demonstration Context #2: Civilian autonomous expressway driving (commercial automotive)
        - Scope suitable to assess SE MPT strengths, limitations, and potential for transition into use
    - Document further SE research issues in AES development, verification and validation
        - V&V needs, issues and approaches
        - Refinements and extensions to the SE MPT and process model for AES requirements
        - Refinements and extensions to the AES cognitive architecture model
        - User-friendly software to implement the MPT
4. Plan of Work
    - Apply the SE AES behavior requirements MPT in two trial application contexts
        - Military convoy and individual civilian
    - Review the execution and results with potential end users and transition partners
    - Document the project, results, findings and recommendations
5. Transition assessment and co-development partners
    - TARDEC Autonomous Convoy Project
        - Ongoing TARDEC S&T project
        - Was the case study for RT-148 on "SE Workflow In Model-Based Engineering," coinciding with the inception of the S&T project
        - Additional TARDEC funding is not available at this time
        - Potential for a collaborative follow-on project through TARDEC's Automotive Research Center
    - Level-4 Autonomy Project with a Tier-1 Automotive Supplier (Hyundai-Mobis)

- Developing and piloting course materials for SE for autonomy-enabled vehicles Spring-Summer 2018
- Pilot application to Level-4 autonomy capability demonstration Spring-Summer 2019
- WSU Academic Initiatives
- Graduate course in SE for autonomy-enabled vehicles to be offered at WSU Fall 2018, based on SERC RT and commercial pilot

- Other external coordination opportunities
  - Early Synthetic Prototyping at TARDEC
  - Autonomy TEV&V Community of Interest

6. Risks, Mitigations and Opportunities
   - Choice of software platform to demonstrate the AES cognitive architecture model
     - Choice of software platform for BDI agent modeling could (1) limit ability to implement the AES cognitive architecture model, and/or (2) increase time and effort to use
       - Ease-of-use, compatibility with the conceptual AES cognitive architecture model, development, analysis and simulation support, potential for transition to real-time distributed embedded cyber-physical systems
       - General agent modeling with hybrid state-flow machines, e.g. Matlab Simulink/Stateflow, Ptolemy II
       - Agent modeling with specific BDI implementations, e.g., JADEX, JACK
       - Commercial (Matlab Simulink/Stateflow, JACK) vs mature non-commercial platforms (Ptolemy II, JADEX)
     - Implementation for R&D proof-of-concept is a demonstration risk, but does not have long term consequences
     - Mitigation is to accept the risk
   - SE for Heuristic Machine Learning (HML) in AES
     - HML (Convolutional Neural Nets, Deep Learning, Genetic Algorithms, etc.) promise great potential capability, but SE methods to design, train, and test the methods, and then to assure confidence in capability and scope of applicability are lacking
     - DoD SE lacks MPT to identify the high-value, low-risk roles of HML in DoD systems
     - Calibrated trust: risks of over-expectations, under confidence and misunderstanding
     - Technical challenge A: data sets for factor selection, training and assessment – statistical sufficiency, robustness, bias, and skewness considerations
     - Technical challenge B: V&V of the engineering process for HML
     - Mitigation is to declare SE for HML to be out of scope
   - Verification & Validation of AES Requirements and Specifications
     - Development of a behavior requirements framework that supports and enables early and continuous V&V is high priority for assurance that "we are building the right thing"
     - V&V MPT for AES and AES requirements is largely *terra incognita*, and a challenge beyond the time and resources of this project
     - Mitigate the risk by identifying V&V issues and opportunities, but stop short addressing V&V MPT

   Dr. Witus has also been leading several complementary activities:
   - He is conducting the SE pilot study for Level-4 autonomous driving with Hyundai-Mobis
   - He is advising a TARDEC ground systems engineer as a WSU Ph.D. student in addressing key research activities

- He is developing WSU's planned Fall 2018 course in SE for Autonomous Vehicles

APPENDIX A: LIST OF PUBLICATIONS RESULTED

Rosa, Wilson, et al. "Early phase cost models for agile software processes in the US DoD." Empirical Software Engineering and Measurement (ESEM), 2017 ACM/IEEE International Symposium on. IEEE, 2017.

Barry Boehm, Life Cycle Resilience Depends on Maintainability, GSAW 2018 Tutorial Abstract

Barry Boehm, Pooyan Behnamghader, Non-Technical Sources of Technical Debt and the Software Maintenance Readiness Framework (SMRF), submitted to ICSME 2018.

Hall, Robert, “Utilizing A Model-Based Systems Engineering Approach to Develop a Combat System Product Line.

APPENDIX B: CITED AND RELATED REFERENCES

Abdallah, N. B., Mouhous-Voyneau, N., & Denoeux, T. (2013, July). Using Dempster-Shafer Theory to model uncertainty in climate change and environmental impact assessments. In Information Fusion (FUSION), 2013 16th International Conference on (pp. 2117-2124). IEEE.

Agarwal, H., Renaud, J. E., Preston, E. L., & Padmanabhan, D. (2004). Uncertainty quantification using evidence theory in multidisciplinary design optimization. Reliability Engineering & System Safety, 85(1-3), 281-294.

Einhorn, H. J., and Hogarth, R. M. (1985). Ambiguity and uncertainty in probabilistic inference. Psychological review, 92(4), 433.

Foley, B. G. (2012). A Dempster-Shafer Method for Multi-Sensor Fusion. MSc thesis, Department of Mathematics and Statistics, Air Force Institute of Technology

Sitterle, V. B., Brimhall, E. L., Freeman, D. F., Balestrini-Robinson, S., Ender, T. R., & Goerger, S. R. (2017). Bringing Operational Perspectives into the Analysis of Engineered Resilient Systems. INSIGHT, 20(3), 47-55.

Wilkening, D. A. (2000). A simple model for calculating ballistic missile defense effectiveness. Science & Global Security, 8(2), 183-215.

Wu, H., Siegel, M., Stiefelhagen, R., & Yang, J. (2002). Sensor fusion using Dempster-Shafer theory [for context-aware HCI]. In Instrumentation and Measurement Technology Conference, 2002. IMTC/2002. Proceedings of the 19th IEEE(Vol. 1, pp. 7-12). IEEE.

Xu, D. L. (2012). An introduction and survey of the evidential reasoning approach for multiple criteria decision analysis. Annals of Operations Research, 195(1), 163-187.

Zadeh, L. A. (1986). A simple view of the Dempster-Shafer theory of evidence and its implication for the rule of combination. AI magazine, 7(2), 85.

Zimmermann, H. J. (2000). An application-oriented view of modeling uncertainty. European Journal of operational research, 122(2), 190-198.

de Weck, O.L., Ross, A.M., and Rhodes, D.H. (2012). “Investigating Relationships and Semantic Sets amongst System Lifecycle Properties (Ilities)”. *3rd International Engineering Systems Symposium*. CESUN 2012. TU Delft. 18-20 June 2012

Lockett, J., Swan, M., Unai, K., (2017). The Agile Systems Framework: Enterprise Content Management Case. 15th Annual Conference on Systems Engineering Research. Redondo Beach, CA.

Ross, A.M., Beesemyer, J.C., and Rhodes, D.H., (2011) "A Prescriptive Semantic Basis for System Lifecycle Properties", SEAri Working Paper WP-2011-2-1. MIT. Cambridge, MA. http://seari.mit.edu/papers.php.

Ross, A.M., Rhodes, D.H., and Hastings, D.E. (2008), "Defining Changeability: Reconciling Flexibility. Adaptability, Scalability, Modifiability, and Robustness for Maintaining Lifecycle Value," *Systems Engineering*. Vol. 11. No. 3. pp. 246-262

Ross, A.M., and Rhodes, D.H. (2015). "Towards a Prescriptive Semantic Basis for Change-type Ilities." 13th Conference on Systems Engineering Research. Hoboken, NJ. March 2015