



2019 - New Project Incubator

Technical Report SERC-2019-TR-017

December 2019

Principal Investigators:

Mr. Thomas McDermott, Stevens Institute of Technology

Dr. Dinesh Verma, Stevens Institute of Technology

Co-PIs:

Dr. Mark Austin, University of Maryland

Dr. Paul Collopy, University of Alabama Huntsville

Dr. Laura Freeman, Virginia Tech

Dr. Richard Malak, Texas A&M University

Dr. Michael Pennock, Stevens Institute of Technology

Dr. Daniel Selva, Texas A&M University

Dr. Valerie Sitterle, Georgia Tech Research Institute

Dr. Eric Weisel, Old Dominion University

Dr. Ye Yang, Stevens Institute of Technology

Sponsor: OUSD(RE)

Copyright © 2019 Stevens Institute of Technology, Systems Engineering Research Center

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 [HQ0034-19-D-0003, TO#0330].

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).

No Warranty.

This Stevens Institute of Technology and Systems Engineering Research Center Material is furnished on an “as-is” basis. Stevens Institute of Technology makes no warranties of any kind, either expressed or implied, as to any matter including, but not limited to, warranty of fitness for purpose or merchantability, exclusivity, or results obtained from use of the material. Stevens Institute of Technology does not make any warranty of any kind with respect to freedom from patent, trademark, or copyright infringement.

This material has been approved for public release and unlimited distribution.

TABLE OF CONTENTS

Introduction	1
1. Models for Efficient Testing (MET) – Dr. Ye Yang, Stevens Institute of Technology	6
1.1 Introduction and Motivation	6
1.2 Current Practices and Its Limitation	7
1.3 Proposed Research And Potential Impact.....	10
1.4 Significance and Expected Contributions	17
1.5 Risks and Mitigation Plans.....	18
1.6 Budget	18
1.7 References.....	18
2. Explaining Model Composability using Causal Graphs– Dr. Michael Pennock, Stevens Institute of Technology.....	20
2.1 Problem Statement, Motivations, and Objectives	20
2.2 Current Practice and Its Limitations	21
2.3 Research Approach	22
2.4 Results of the Incubator Study	26
2.5 Proposed Work for the Next Phase.....	34
2.6 References.....	35
3. Test, Evaluation, Verification and Validation (V&V and T&E) Competency Framework – Dr. Laura Freeman, Virginia Tech	37
3.1 Overview and Motivation.....	37
3.2 Current State of V&V and T&E	40
3.3 Research Plan	44
3.4 Execution Strategy and Deliverables.....	51
3.5 References.....	53
4. Validation of AI-Enabled and Autonomous Learning Systems – Dr. Paul Collopy, University of Alabama Huntsville and Dr. Valerie Sitterle, Georgia Tech Research Institute	54
4.1 Objectives of the Research.....	54
4.2 Current Practice and its Limitations	55
4.3 Research Approach	63
4.4 Relevance of the Research	71
4.5 Risks, Mitigations, and Payoffs.....	72
4.6 Budget	73
4.7 Timeline	73
4.8 Project Evaluation and Measurement.....	73
4.9 References.....	73
5. Fostering Human Learning from Cognitive Assistants for Design Space Exploration – Dr. Daniel Selva, Texas A&M University	76
5.1 Executive Summary	76
5.2 Background and Motivation.....	76
5.3 Research Objectives	81
5.4 Technical Approach	81
5.5 Preliminary Work Performed in Incubator Phase	84

5.6 Proposed Research	86
5.7 Management Plan	87
5.8 References	88
6. Architecting Digital Twins for Model-Centric Engineering: A Combined Semantic Modeling and Machine Learning Approach – Dr. Mark Austin, University of Maryland	89
6.1 Background.....	89
6.2 Proposed Work for Post Incubator Research.....	91
6.3 Accomplishment Overview – Progress Towards Objectives	96
6.4 Agenda and Budget for Proposed Research.....	103
6.5 References.....	104
7. Ecology-Inspired Technique for Resilient Engineered System of Systems Design – Dr. Richard Malak, Texas A&M University	105
Abstract	105
7.1 Introduction.....	105
7.2 ENA, the degree of system order, and the ecological fitness of function.....	107
7.3 Investigating the <i>fitness</i> trends in a notional hostiles surveillance SoS.....	110
7.4 Results and Discussion	113
7.5 Conclusions.....	116
7.6 Future Work	116
7.7 References.....	118
8. Quantifying Mission Impact for Technology Alternatives – Dr. Eric Weisel, Old Dominion University.....	120
8.1 Problem Statement, Motivation, and Objectives.....	120
8.2 Basis for the Research Opportunity	120
8.3 Technical Approach	121
8.4 2019 Incubator Program Research Future Challenge Task	122
8.5 Research Team Qualifications – Planning for Success	129
8.6 Relevance of the Research	129
8.7 Risk Assessment and Mitigation Plan.....	130
8.8 Project Cost, Schedule and Milestones	130
8.9 References.....	131

INTRODUCTION

The need for new project incubation is an integral part of the Systems Engineering Research Center (SERC) 2019-2023 technical plan as there continues to be a significant need to identify and cultivate new ideas that may become critical research programs that address emerging challenges. The SERC performs research on 18-20 active tasks on well-defined topics that are aligned with the SERC's research strategy. While this research has great potential to have a transformative impact on systems engineering in both DoD and the intelligence community, it is also important to identify and nurture new ideas that may become the critical research programs for emerging challenges.

The open proposal call period was March 14-May 1, 2019 with the objective of identifying and developing several short white papers outlining research programs with a significant potential to improve the practice of systems engineering. A total of 58 responses were received.

Table 1: Responses to 2019 SERC Incubator Grant Solicitation

University	PI	Team	Title	Research Area:
University of Maryland, SIT	Mark Austin	Maria Coelho, Mark Blackburn	Architecting Digital Twins for Model-Centric Engineering: A Combined Semantic Modeling and Machine Learning Approach	AI
AOE, VT	Seongim Choi	Xi Chen	Real-time, Dynamic Operation of Human-Robot Team using Sliding and Blending Autonomy Policies	AI
VT	Bill Clark		RFML Kraken	AI
UAH	David Coe	Jeffrey H. Kulick, Aleksandar Milenkovic, Letha Etzkorn	Architectural Frameworks for Establishing Trust in Autonomous Cyber-Physical Systems	AI
VT	Xinwei Deng	Yili Hong, Laura Freeman	Robustness and Fair Evaluation for Data-Driven AI	AI
Dartmouth	Lance Fiondella		AI & Autonomy: A Common Weakness Enumeration to support Failure Modes and Effects Criticality Analysis of Machine Learning-augmented Systems	AI
Missouri University of Science and Technology	Abhijit Gosavi	Lesley Sneed	DRL-FA-SoS: A Deep Reinforcement Learning Approach to Flexible Architectures in a System of Systems	AI
SIT	Yeganeh M. Hayeri	Saeed Vasebi, Christopher Caligara	Recurrent neural network based group feature selection for time series prediction: Case study for preventive maintenance of aircraft engines	AI

VT	William C. Headley		Radio Frequency Machine Learning Intuition through Virtual Reality Visualization Tools	AI
VT	Thomas Krauss		Vive Lighthouse Indoor Localization	AI
VT	Nathan Lau	Ryan K. Williams	Trust Through Competition: Mixed Cooperative-Competitive Strategies in Human-Machine Decision-Making	AI
Wayne State University	Yanchao Liu		Combining Trade Space Analysis and Mathematical Programming Approaches for Vehicle Lightweighting Optimization	AI
Wayne State University	Yanchao Liu		Drone Swarm Systems Engineering for Ship and Critical Infrastructure Inspection	AI
USC	Azad M. Madni	Dr. D. Erwin (Co-I), Dr. M. Sievers (Senior Scientist), P. Pouya (RA)	Closed Loop Intelligent Risk Management Using Model-Referenced Methods	AI
USC	Azad M. Madni	Dr. D. Erwin (Co-I), Dr. Mike Sievers, Shatad Purohit	Semi-automated Architectural Decomposition and Analysis Using AI-Based Methods	AI
Purdue	Karen Marais	Milind Kulkarni	Model Mining: Learning Models of Safety-Critical Human-in-the-Loop Systems	AI
VT	Alan J. Michael	Lauren Wong	Machine Learning Resiliency on Space-based Platforms	AI
UAH	Leonard Petnga		Detecting Faults and Reasoning about System of Systems Failure (Sos - Daref)	AI
Purdue	Ali K. Raz		Building Autonomy via Systems Engineering (BASE)	AI
Texas A&M	Daniel Selva		Fostering human learning from cognitive assistants for design space exploration	AI
UAH, GTRI	Paul Collopy (UAH), Valerie Sitterle (GTRI)	Jennifer Stevens (UAH, NASA.MSFC), AI SME (GTRI)	Validation of AI-Enabled and Autonomous Systems	AI
SIT	Lu Xiao		Discovering System Performance Issue Taxonomy and Tactics Using AI	AI
VT	Xiaowei Yue	James Kong	Dynamic Data Fusion for Complex Systems in Austere Operating Environments	AI
VT	Xiaowei Yue		Machine Learning for Battlefield Health Care	AI
Wayne State University	Celestine Aguwa		Human Capital Development through the Integration of STEM Concept with Systems Engineering	HCD
VT	Laura Freeman	Alejandro Salado	Verification & Validation (V&V) / Test & Evaluation (T&E) Competencies	HCD
USC	Azad M. Madni	Dan Erwin, Ayesha Madni	Exploiting Digital Twin Technology During Hands-On Learning of Systems Engineering	HCD

UAH	Bryan Mesmer	Jonathan Patrick	Category and Set Theory for the Creation of Positions and Role Assignments for Systems Engineers	HCD
Wayne State University	Jeremy L. Rickli		Production Systems Engineering for Advanced Technology Systems with Digital Engineering and Manufacturing Processes	HCD
VT	Kostas Triantas	Navid Ghaffarzadegan, Jake Grohs, Niyousha Hosseinichimeh, David Knight	Understanding Complexity of Engineered Systems	HCD
Purdue	Ilias Bilionis		UNDERSTANDING COST OVERRUNS IN MAJOR GOVERNMENT-FUNDED ACQUISITIONS	ME
VT	Bradley Davis		Adapting Antennas for Complex Environments	ME
PSU	Nil Ergin	Satish Srinivasan, Raghu Sangwan, Colin Neill	Ranking Critical Activities in Process Architectures	ME
SIT	Steven Hoffenson and Zhongyuan Yu		Decision Analysis and Support for Heterogeneous Engineering Systems (DASHES)	ME
Texas A&M	Richard Malak	Astrid Layton	Ecology-Inspired Technique for Resilience Evaluation of Engineered Systems of Systems	ME
UAH	Bryan Mesmer	Kristin Weger, Joy Robinson	Modeling incentive schemes that overcome mission enterprise fragmentation	ME
SIT	Michael Pennock		Explaining Model Composability using Causal Graphs	ME
VT	Alejandro Salado	Hanumanthrao Kannan	Modeling the Art of Systems Architecture: Formally Coupling Architecture and Organization in Model-Based Systems Engineering (MBSE) Environments	ME
VT	Alejandro Salado		Predicting Emergent System Behavior by Coupling Behavioral Models	ME
VT	Subhash Sarin	Manish Bansal	Streamlining the Army Sustainable Readiness Scheduling System	ME
ODU	Eric Weisel	Erika Frydenlund, Peter Foytik	Quantifying Mission Impact for Technology Alternatives	ME
VT	Deborah E. Dickerson (nee Young-Corbett)		Prevention through Design: Innovative Framework for Risk Mitigation	TS
GTRI	J. Charles Domercant	Michael Shearin, Sean Higgins, Richard Wise, Cyber-security (GTRI)	Formal MBSE Architecture Expressions for Cyber-Security	TS
PSU, US Army Corps of Engineers, VT,	Darryl Farber	Michael Deegan, Ken Harmon, Conrad Tucker, John Hodgson, Azad Madni	Engineering Systems Analysis of the Resilience of The U.S. Inland Rivers, Ports, and Terminals System	TS

Carnegie Mellon, USC				
University of South Florida	Yao Liu		Combating Denial-of-Service Attacks in the Era of Internet of Things	TS
USC	Nenad Medvidovic		Identifying Software Vulnerability Patterns and Developing Patches for Them	TS
VT	Alan J. Michaels	Joey McCollum	Positive Document Control	TS
UAH	Biswajit Ray	Aleksandar Milenkovic	In-Memory Security Primitives for Cyber Physical Systems	TS
ODU	Sachin Shetty		Optimal Cyber Defense Remediation by Balancing Resilience and Safety	TS
USC	Pooyan Behnamghader		Highly Scalable Cloud-Based Analysis of Mission-Critical Software Quality	VEL
VT	Joseph D. Gaeddert		Accelerating Digital Phased-Array Prototyping in Support of Cyber Spectral Awareness	VEL
VT	Robert Gramacy	Dave Higdon, Laura Freeman	Digital Test and Evaluation (T&E)	VEL
SIT	Paul T. Grogan and Steven Hoffenson		Model-based Analysis, Inference, and Synthesis Tools for Responsive Engineering (MAISTRE)	VEL
VT	Hanumanthrao Kannan	Benjamin C. Jantzen	A Formal Logic-Based Approach to Capturing Knowledge in the Development of Large-Scale Complex Engineered Systems	VEL
VT	Devin Ridge		HDL Core Portability Between FPGA Platforms Utilizing OpenCPI	VEL
VT	Kostas Triantis		A Modeling Approach for Engineering Resilient Systems	VEL
SIT	Ye Yang		Models for Efficient Testing (MET)	VEL
Wayne State University	Murat Yildirim		Secure and Scalable Scheduling in Condition-Based Maintenance Plus: A System-of-Systems Framework for Large Scale Military Fleets	VEL

Each of the received proposals were reviewed by the Research Council except where they noted a potential conflict of interest. Each member provided a final score based on an equal weighing in each of the following four criteria as well as a set of short comments:

- Intellectual Merit
- Clarity of Vision
- Past Performance
- Potential Strategic Impact

The sponsor independently reviewed the projects based on these criteria then met with the PI to discuss final selections. In this case, the selected eight proposals were favorably reviewed by the Research Council. It was believed by the sponsor that interviews were not necessary to make the final determination of the awards.

Of these, the following eight proposals were selected:

University	PI	Title	Research Area
SIT	Ye Yang	Models for Efficient Testing (MET)	VEL
SIT	Michael Pennock	Explaining Model Composability using Causal Graphs	ME
VT	Laura Freeman	Verification & Validation (V&V) / Test & Evaluation (T&E) Competencies	HCD
UAH, GTRI	Paul Collopy (UAH), Valerie Sitterle (GTRI)	Validation of AI-Enabled and Autonomous Systems	AI
Texas A&M	Daniel Selva	Fostering human learning from cognitive assistants for design space exploration	AI
University of Maryland	Mark Austin	Architecting Digital Twins for Model-Centric Engineering: A Combined Semantic Modeling and Machine Learning Approach	AI
Texas A&M	Richard Malak	Ecology-Inspired Technique for Resilience Evaluation of Engineered Systems of Systems	ME
ODU	Eric Weisel	Quantifying Mission Impact for Technology Alternatives	ME

This report contains the white papers that was supported by each of these eight funded proposals.

1. MODELS FOR EFFICIENT TESTING (MET) – DR. YE YANG, STEVENS INSTITUTE OF TECHNOLOGY

1.1 INTRODUCTION AND MOTIVATION

Trade-offs such as “how much testing is enough” are critical yet challenging decisions in planning and managing software and systems testing [1, 2]. Insufficient testing can lead to unsatisfying product quality, while excessive testing can result in potential schedule delays and low cost-effectiveness. This is especially true for the software-intensive systems in the defense domain, given their ever increasing system complexity, the lack of testing processes and resources, as well as the intense schedule pressure.

This project is motivated by the findings reported from the recent SERC workshop on “Model-Based System Assurance (MBSA)” [3], which relate to the model-based testing practices across the US government. It states that “there is no formal testing process across different systems across the government. Ad hoc, manual, decentralized testing activities make system assurance testing very challenging.” Consequently, when it comes for all testers to answer the typical question as to “when to stop testing”, there is lack of analytical or scientific methods to back such decision making. Most common criteria practiced in the software and system testing are either deadline-driven or coverage-driven. The former refers to the dependence on committed / planned testing deadlines which are about to expire. The latter refers to the satisfaction of certain test coverage target, e.g. after execution of all the planned test cases. It is obvious that neither case is a direct representation for the quality of the system under test.

By definition, the purpose of testing is to discover the existence of defects, not the lack of defects. Our early investigation on real-world testing data reveals that there are large variations in bug arrival rate of testing tasks, and in task’s duration and consumed cost for achieving the same quality level. It is very challenging for managers to come up with reasonable experience-based decisions, and there is an average of 32% wasteful testing spending in our previous empirical investigation [4]. We argue that the limitations lie in the lack of processes, and more fundamentally, the lack of metrics and methods to support dynamic measurement and process visibility among distributed testing teams, in order to support better planning and managing testing tasks. For example, the Rayleigh curve is a classical model for predicting the dynamic defect arrival in software testing and reliability engineering [6], however, it requires a fundamental measurement model to be effectively applied.

With the rapid shift to agile development practices, project and testing managers are facing tremendous decisions impacting their daily actions, e.g. deciding how to better allocate scarce testing resources among competing testing tasks, monitoring and evaluating testing sufficiency of a large number of test cases. In order to improve the testing efficiency and cost-effectiveness, there is a compelling need, as suggested in the SERC MBSA report [3], to conduct more exploratory research on developing “new metrics and methods for evaluating of the credibility of ... testing data and analysis results.” Such models need to be time-sensitive, context-sensitive, and non-interruptive with respect to testing manager’s typical decision scenarios.

In this project, we plan to investigate and develop a set of models to support efficient testing management, named MET, and particularly to support two primary types of testing decisions: matching distributed testing resources to tasks and predicting testing completion status. More specifically, the research objectives are three-folds: 1) to develop a testing measurement model and metrics for planning and measuring distributed testing processes; 2) to develop a machine-learning based model for learning potential defect detection capability of distributed testing teams and recommending optimal team formations to satisfy specific testing objectives for a new testing task, e.g. maximizing the defect detection efficiency and the testing effectiveness; and 3) to develop a machine-learning based model to dynamically monitor, aggregate, and predict testing progress towards completion among distributed testing teams. The ultimate goals for this research is to enable actionable, value-driven decision making on resource allocation and utilization faced by testing managers.

1.2 CURRENT PRACTICES AND ITS LIMITATION

In general software testing practice, testing managers prepare the testing tasks (including the software under test and test requirements), and distribute to certain distributed testing teams, as shown in Figure 1. Each testing team/worker then executes the testing task and submits testing reports, typically summarizing test input, test steps, test results, etc. The testing managers need to monitor testing reports, inspect and verify each

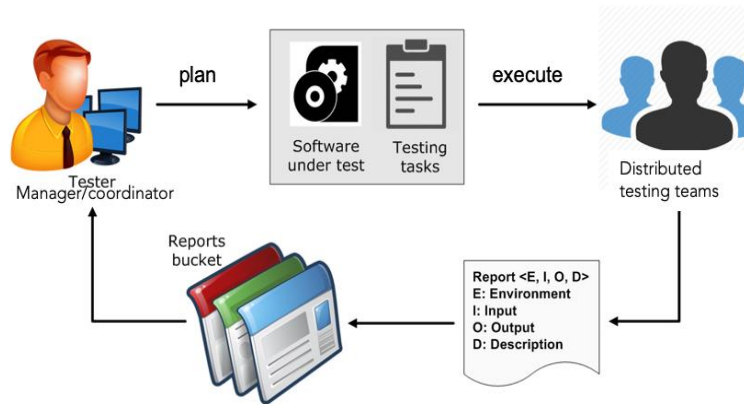


Fig. 1 A Typical Testing Process

report for their tasks manually or using automatic tool support. Generally, each report will be characterized using two attributes: 1) whether it contains a valid bug; 2) if yes, whether it is a duplicate bug that has been previously reported in other reports. In the following paper, if not specified, when we say “bug” or “unique bug”, we mean the corresponding report contains a bug and the bug is not the duplicate of previously submitted ones.

Testing managers typically plan for the close of testing tasks based on organization’s established testing criterion, or expert experience if such criteria is non-existing. Well known testing criteria commonly used in software testing are statement coverage and branch coverage [7, 34, 35]. The rationale is that the higher the coverage, the higher the chances of catching a code feature that causes a failure. While this rationale is intuitive, it relies on an important assumption which leads to a common problem regarding the quality of the testing oracle [8]. If the oracle does not suffice to cover the error, then there is no way to detect the faults even when coverage criteria reaches to 100%.

In recent years, crowdtesting has become an effective alternative and supplementary to traditional testing, especially for exploratory testing on mobile applications [9-13]. It entrusts testing tasks to online, distributed, testers whose diverse testing environments, background, and skill sets could significantly contribute to more reliable, cost-effective, and efficient testing results [12, 13]. Crowdtesting has been adopted by many software organizations, including but not limited to Google, Facebook, Amazon, Microsoft [14, 15]. Crowdtesting managers usually set up either a fixed period (e.g., 5 days) or a fixed number of participant (e.g., recruiting 400 crowd workers) as the close criteria. If either of the criteria is met first, then the task will be automatically closed. There are different payout schema in crowdtesting, e.g., pay by report [4]. Generally, the cost of a task is positively correlated with the number of received reports, thus with the close time.

In Crowdtesting marketplace, software testing workers are shared resources and typically independent of task requesting organizations. Various incentive factors are reported including gaining skills, getting feedback, making friends, earning money, having fun, getting peer recognition and getting sense of accomplishment [16]. We've found that open call tasks frequently lead to ad hoc worker behaviors [4, 5]. For example, in some cases, testers may choose tasks they are not good at, and end up with no bug detected [4] or task failure [5]; in other cases, many testers with similar experience may submit many duplicate bugs and result in waste of effort [4].

Fig. 2 shows typical bug arrival curves in 9 mobile application testing projects. While the bug arrival curves differ, somewhat, note that they all exhibit the same "plateau effect", after which (i.e., red point in each project) new testing reports contains no new bugs. We define optimal close time as the time when the last report is received after when there are no new bugs submitted. This has led to the development of iSENSE for testing completion prediction [4]. However, iSENSE comes with two limitations. One is its reliance on the manual labeling of duplicate bugs, in maintaining the Defect Lookup Table. The other is its false-alarm corresponding to the Stay-Rise pattern of the defect arrival curves (as shown in the last 3 projects in Fig. 2, many new defects discovered after the blue dots).

These limitations and emergent crowdtesting paradigm drive the motivation behind this project, and provide opportunities for developing in-depth understanding and characterizing distributed software testing workers, learning successful knowledge of team formation and testing process management from historical projects, in order to improve efficiency and effectiveness distributed testing processes and resources.

Specifically, we leverage testing worker data extracted from crowdtesting marketplace, develop models representing testing context, knowledge, as well as progress, and train machine learning models to support dynamic, in-process decision making on testing team formation and termination automation. We believe that the complexity of mobile applications and the unpredictability of distributed crowdtesting processes can serve as an interesting representation for comparable complex systems in defense domain, as well as general testing resource's expertise, bug detection capability, and behavior patterns. Such models also help to address existing challenges in lack of visibility into the black box crowdtesting process and its actual progress. The proposed research will benefit both traditional and crowdtesting process modeling, measurement, and management practices.

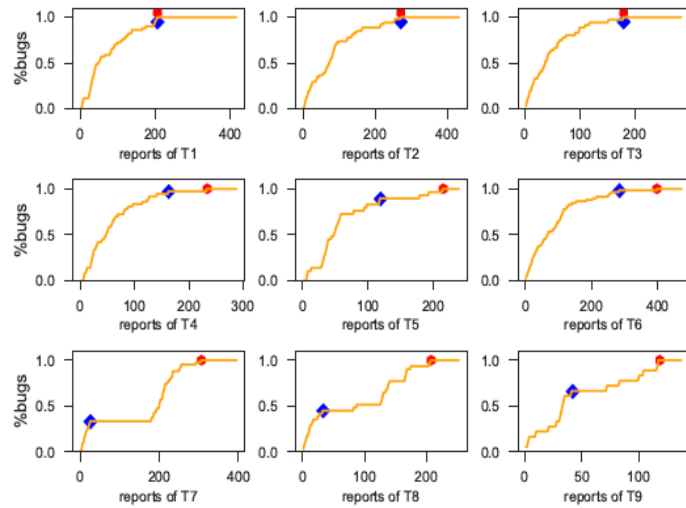


Fig. 2 Three Dominant Categories of Defect Arrival Curves

1.3 PROPOSED RESEARCH AND POTENTIAL IMPACT

In this project, we propose to develop the MET framework, which stands for Models for Efficient Testing. MET consists of a set of measurement models, metrics, and machine-learning based models for planning and managing efficient testing processes among distributed testing teams. It consists of three main components, as illustrated in Figure 3:

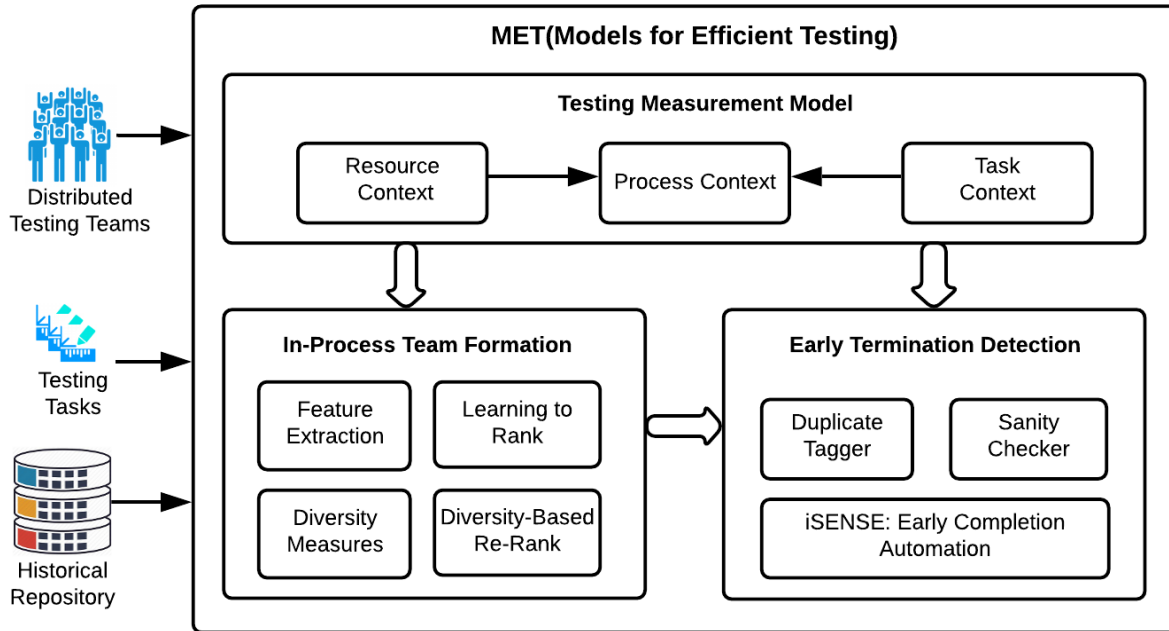


Fig. 3 Overview of MET Research Framework

- A testing measurement model as well as metrics for characterizing the representative contextual factors of a testing task, the configuration profiles of distributed testing resource, and the dynamic status of testing progress;
- A testing team formation model leveraging various machine-learning algorithms for matching, learning, ranking, and dynamically tuning the configuration of distributed testing teams to maximize testing adequacy, leveraging natural language processing (NLP) and learning-to-rank algorithms
- A testing completion prediction model employing incremental sampling technique to dynamically monitor, aggregate testing report data from distributed teams, and measure testing progress towards completion.

More details on our proposed approaches are described next.

1.3.1 Testing Measurement Model

The Testing Measurement Model (TMM) characterizes the time-sensitive, contextual information from three perspectives, i.e., the Task Context, the Process Context, and the Resource Context, respectively, as shown in Figure 4. Contextualization is an important set to enable the characterizing, learning, and undercover of underlying patterns of software testing processes. TMM is constructed towards that purpose by characterizing the in-process testing requirements, software testing progress, and dynamic situation of distributed testing teams at a finely-defined granularity.

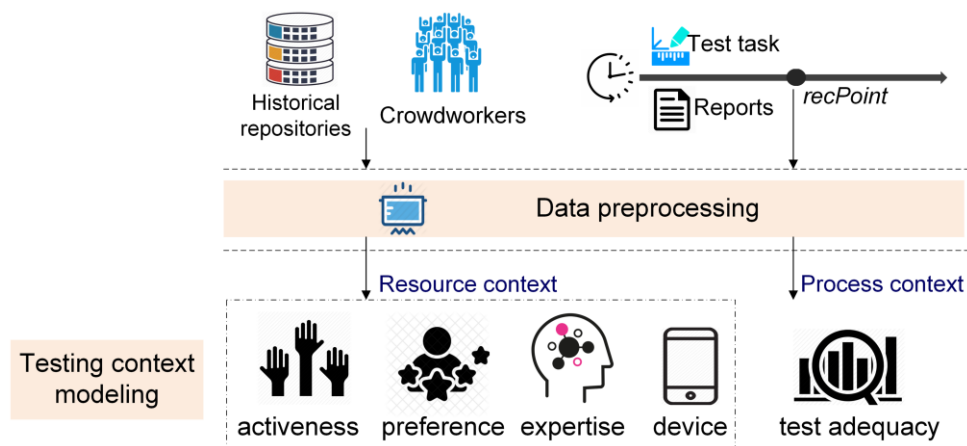


Fig. 4. Testing Measurement Model

1.3.1.1 Task Context

The task context is characterizing the product factors of a testing task. This project proposes to represent the task requirement document in the vector space of descriptive terms list and denote it as task terms vector. The main reason is to be able to leverage natural language processing techniques and ensure the task context model independent of different crowdsourcing tasks as well as different domains. This way, the proposed approach is applicable to supporting different tasks types through tailorable, more elaborated descriptive terms to represent specialty skillsets in task context model, which could potentially have minor impact on the applicability of the other learning and ranking components to be introduced in later subsections. To model task context, each task requirements document goes through standard word segmentation, stop-word removal, with synonym replacement being applied to reduce noise. As an output, each document is represented using a vector of terms. Since some high-frequency terms as well as some other low-frequency terms both correspond to less predictive and contribute less in modeling the task context, in terms of describing requirements/constraints from functional, performance, security, technical, platform, as well as submission guideline perspectives. Therefore, this project will construct a descriptive terms list to facilitate the effective modeling and tuning the task context model. The basic idea is to first pre-process all the documents in the training data set and obtain the terms of each document. Then, the terms are ranked according to the number of documents in which a term appears (i.e., document frequency, also known as df), and filter out 5% terms with the highest document frequency and

5% terms with the lowest document frequency (i.e., less predictive terms) following previous work [18, 20]. Note that, since the documents in software crowdsourcing tasks are typical short in length, the term frequency (also known as tf), which is another commonly-used metric in information retrieval [21], is not discriminative, so this project will only use document frequency to rank the terms. Finally, derive the final descriptive terms list and use to represent each document in the vector space of the descriptive terms.

1.3.1.2 Process Context

This project proposes to model the testing process context using the following notions:

- **Set of Testing Resources:** A dynamically formed, distributed testing team consisting of all testers engaged at the task at a specific point of time.
- **Set of Testing Reports:** The set of received testing reports, represented in the matrix of descriptive term.
- **Set of Unique Bugs:** The set of received unique, represented in the matrix of descriptive term. This notion will be elaborated with respect to specific types of testing tasks.
- **Testing Adequacy:** A measure of the testing progress regarding to what degree each descriptive term, t_j , of testing requirements (i.e., task terms vector) has been tested. It is measured using the ratio between the number of testing reports with t_j and all submitted testing reports received at that point of time. This is a probability measured in a dynamic and iterative manner, regarding to what degree each descriptive term of task requirements has been sufficiently covered with respect to dynamic process context indicating testing progress at the time.

$$TestAdeq(t_j) = \frac{\text{number of bug reports with } t_j}{\text{number of received bug reports in a task}} \quad (1)$$

where $t_j \in \text{task terms vector}$. The larger $TestAdeq(t_j)$, the more adequate of testing for the corresponding aspects of the task. This definition enables the learning of underlying knowledge to match testers' expertise or preference with inadequate-tested terms at a finer granularity.

1.3.1.3 Resource Context

Our preliminary data analysis explores the characteristics of software testing teams which can influence their test participation and bug detection [17]. As shown in Figure 5, the distribution of crowdworkers' activity intensity (left chart) and technique preferences/expertise (right chart) vary dramatically from time to time.

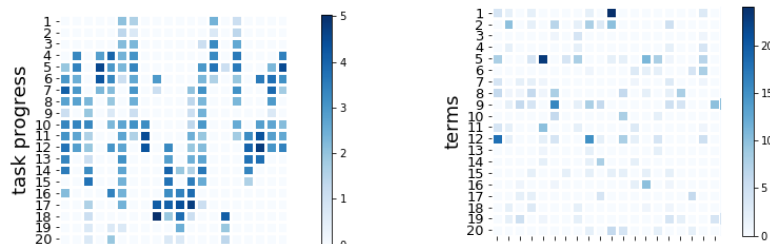


Fig. 5 Empirical Observation of Testers Activeness and Expertise

The x-axis is the random-selected 20 testers among the top-50 testers ranked by the number of submitted reports, and the y-axis is 20 equal-sized time interval which is obtained by dividing the whole time space. The darker color to denote a worker submitting more reports during the specific time interval. It shows that the crowdworkers' activity intensity varies greatly. The differences across columns in the heat map further reveal the diversified preference across workers. Intuitively, if a crowd worker is active and has a preference on the specific aspects of a task, he/she would show greater interest to engage and detect bugs.

Motivated by the above observations, we plan to use three dynamic attributes including *preference*, *expertise*, and *activeness of testers* to model the context of a distributed testing team or community. First, testers can have different **preference** on different domains of testing tasks, i.e., only conduct certain types of tasks; and testers with specific preference on the given tasks would be more likely to conduct the task and then detect bugs. The **preference** of worker w regarding each descriptive term t_j is defined as following equation (2), based on bayes rules [36]. In other words, it is the probability of recommending the worker w when aiming at generating a report with specific term t_j .

$$ProbPref(w, t_j) = P(w|t_j) = \frac{tf(w, t_j)}{\sum_{w_k} tf(w_k, t_j)} \cdot \frac{\sum_{w_k} df(w_k)}{df(w)} \quad (2)$$

Second, similarly, testers have diverse skills in different domains, and the testers equipped with enough **expertise** on the related areas of the given tasks would have large possibility to successfully complete the task, e.g. develop a component with satisfactory quality, or detect bugs. The **expertise** of worker w regarding each descriptive term t_j is defined as following equation (3).

$$ProbExp(w, t_j) = P(w|t_j) = \frac{tf(w, t_j)}{\sum_{w_k} tf(w_k, t_j)} \cdot \frac{\sum_{w_k} df(w_k)}{df(w)} \quad (3)$$

Third, not all testing resources are equally active at the time when recommendation happens; and active testing resources would have large possibility in responding to the task and detecting bugs. **Activeness** of a tester w is characterized using the following four attributes: 1) LastBug: Duration (in hours) between recPoint and the time when worker w 's last bug is submitted; 2) LastReport: Duration (in hours) between recPoint and the time when worker w 's last report is submitted; 3) NumBugs-X: Number of bugs submitted by worker w in past X time, e.g., past 2 weeks; 4) NumReports-X: Number of reports submitted by worker w in past X time, e.g., past 8 hours.

Besides the above three dimensions, this project plans to also include **device of testers** as a separate dimension of resource context. As several studies reported its diversifying role in testing

environment [18, 19]. The resource context model will be designed and evaluated in an iterative manner to effectively capture such changes and uncertainties associated with the tester.

1.3.2 Team Formation Model

The Team Formation Model employs learning-based ranking techniques to iteratively learn the in-process contextual knowledge to match the most appropriate testing teams, i.e., the testers with the greatest potential to detect more unique bugs. The overview of an example in-Process tester recommendation approach is illustrated in Figure 6.

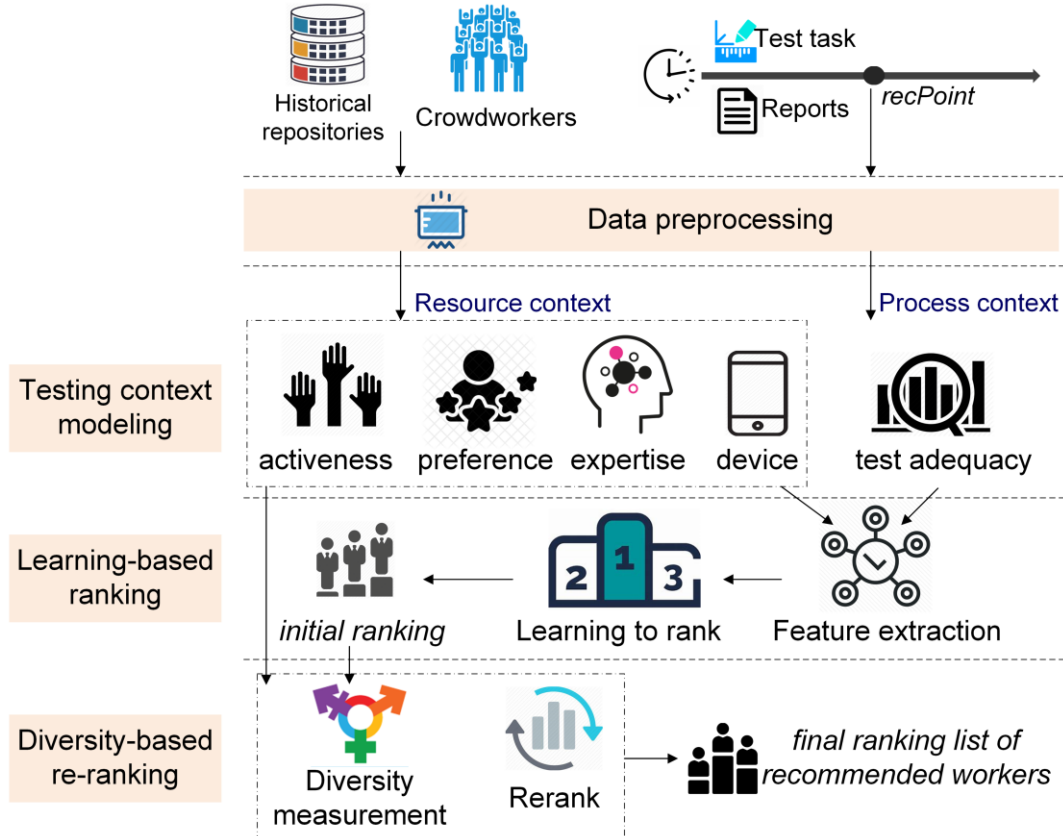


Fig. 6. An In-Process Tester Recommendation Approach

Data Preprocessing and Feature Extraction. Based on the task context model presented in the previous subsection and our previous work, an initial list of 26 features are proposed and will be extracted for training machine learning models, as summarized in Table 1.

Table 1: Examples of Proposed Feature Definition

Category	ID	Feature
Activeness indexing	1	LastBug
	2	LastReport
	3-7	NumBugs-8 hours, NumBugs-24 hours, NumBugs-1 week, NumBugs-2 week, NumBugs- all (i.e., in the past)
	8-12	NumReports-8 hours, NumReports-24 hours, NumReports-1 week, NumReports-2 week, NumReports-all (i.e., in the past)
Preference matching	13-14	Partial-ordered cosine similarity, partial-ordered euclidean similarity between worker's preference and test adequacy
	15-19	Partial-ordered jaccard similarity between worker's preference and test adequacy with the cutoff threshold of 0.0, 0.1, 0.2, 0.3, 0.4
Expertise matching	20-21	Partial-ordered cosine similarity, partial-ordered euclidean similarity between worker's expertise and test adequacy
	22-26	Partial-ordered jaccard similarity between worker's expertise and test adequacy with the cutoff threshold of 0.0, 0.1, 0.2, 0.3, 0.4

Features #1-#12 capture the activeness of a tester. Previous work demonstrated the developer's recent activity has greater indicative effect on his/her future behavior than the activity happened long before [18, 22], so this project will extract the activeness-related features with varying time intervals. Features #13-#19 capture the matching degree between a tester's preference and the inadequate-tested aspects of the task. Features #20-#26 capture the matching degree between the a tester's expertise and the inadequate-tested aspects of the task. The first group of 12 features can be calculated directly based on the activeness attributes defined in the previous section. The second and third group of features are obtained in a similar way by examining the similarities between the descriptive term vector extracted from a tester's past experience and that extracted from the ongoing process context. This project plans to employ three different similarity analysis techniques to produce these two groups of features in a similar way.

Previous work has proven extracting features from different perspectives can help improve the learning performance [23, 24, 25], so the project plans to extract the similarity-related features from different viewpoints. Cosine similarity, euclidean similarity, and jaccard similarity are the three commonly-used similarity measurements and have proven to be efficient in previous researches [26, 27, 28], therefore the project will utilize all these three similarities for feature extraction. In addition, a tester may have extra expertise beyond the task's requirements. In order to alleviate the potential bias introduced by the unrelated expertise, in this project, it is considered using the partial-ordered similarity to constrain the similarity matching only on the descriptive terms within in task terms vector.

Model training. For every task in the training dataset, at each *recPoint*, MET will first obtain the task context, the process context, and the resource context for all candidate testers, then extract the features for each tester. The testers who submitted unique bugs after *recPoint* will be treated as positive instances and labelled as 1. Typically, since there are large number of testers who didn't submit an effective contribution in a specific task, most such data sets will be

unbalanced. Therefore, this project will randomly sample an equal number of testers with the positive instances and label them as 0, following the commonly used one-sided selection under-sampling technique [29].

Ranking based on trained model. MET then apply the trained model to predict and sort the testers based on the predicted potential for detecting unique bugs in a descending order, and treats a ranked list of higher-ranked *recNum* testers (*recNum* is an input parameter since usually only a small set of workers is considered for recommendation) as the output of the learning-based ranking component, i.e., *initial ranking*.

Diversity Based Re-ranking. The Re-ranking algorithm will tune the ranked order according to diversity metrics: **expertise diversity delta** and **device diversity delta**. **Expertise diversity delta** gives higher score to these testers who have most different expertise compared with workers from a task's current tester set. **Device diversity delta** give higher score to those testers who can bring more new device's attributes (e.g., phone type, operating system, IDE environment, etc.) to those of the testers on current re-ranked list, so as to facilitate the exploration in more diverse environment. Example attributes of a worker from historical data sets, i.e., *w's attributes*, include "Samsung SN9009", "Android 4.4.2", "KOT49H.N9009", "WIFI", and so on.

An example implementation of this work has been developed and evaluated on a software crowdtesting dataset. The preliminary results has been submitted ICSE 2020 and is currently under review. In the proposed work, we plan to extend the diversity measures to support cross-disciplinary team configuration which is typically to system testing tasks, e.g. employing the T-shape team configuration [37] to tune the ranking of recommended testing resources in order to balance the depth of technical skills as well as the breadth of multi-disciplinary knowledge areas, in order to detect more bugs efficiently.

1.3.3 Testing Early Completion Model

The Early Completion Model aims at automatically monitoring and detecting early completion status of a testing task in order to avoid wasteful spending on redundant testing effort, as shown in Figure 7. It learns and predicts the appropriate time to terminate a testing task before its planned closing time, as a the testing process is unfolding. In this project, we plan to extend and address two limitations of our previous iSENSE approach [4], which received the ACM SIGSOFT Distinguished Paper Award at 2019 International Conference on Software Engineering (ICSE'19).

To address the first limitation, i.e. excessive redundancy issues in testing tasks, the proposed work will develop an automated duplicate labelling component. This project defines optimal close time for a testing task, as the time when the last unique bug is received. After that time, there is no new bugs submitted except duplicated bug reports. Therefore, any additional testing after that time is not effective and considered as wasteful spending. This project will develop and integrate a duplicate tagger based on semantic analysis into iSENSE, to automatically label the duplicate status of the reports, and the CRC-based (Capture-ReCapture) close estimator is then employed to generate the CRC-based close decision based on the dynamic bug arrival status. This

project plans to adopt word embedding technique to analyze the semantic meaning of testing reports. Word embedding is a feature learning technique in natural language processing where individual words are no longer treated as unique symbols, but represented as d-dimensional vector of real numbers that capture their contextual semantic meanings [30, 31]. Recent studies show that word embedding is effective in detect duplicate status of bug reports in software repositories [32, 33].

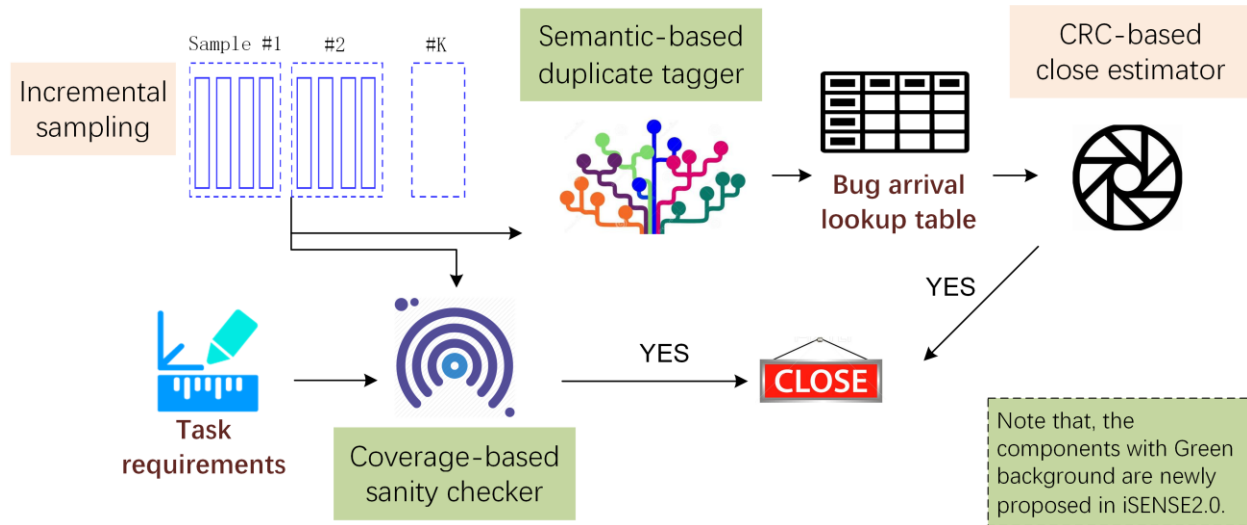


Fig. 7. Overview of Proposed ISENSE 2.0

Another limitation of the iSENSE approach is its performance bottleneck related to the Rise-Stay-Rise bug arrival curve, as illustrated in the last three charts (i.e. Report T7-T9) in Figure 2. The proposed work plans to develop a coverage-based sanity checker to reduce the probability of false alarms and provide improved decision support. Unlike code coverage metrics in traditional testing, e.g., statement coverage, decision coverage, condition coverage, path coverage, etc [34, 35], the project plans to use the metric of Test Adequacy. In mobile application testing context, it is unlikely to obtain source code of the application under test due to confidential considerations, and Test Adequacy will be used to measure the extent to which the task's requirements are explored under distributed testing environment.

1.4 SIGNIFICANCE AND EXPECTED CONTRIBUTIONS

This is the first work to explore in-process learning and recommendation system for distributed testing management. In-process worker recommendation has great potential to facilitate talent identification and utilization for complex, intelligence-intensive tasks. The proposed research integrates concepts and methods from various fields including software engineering, matching theory and machine learning, and develops new models for understand the complicated micro-processes and uncertain testing resources and performances. The evaluation will employ various combined empirical approaches, including real-world historical data sets and case study.

The preliminary evaluation results of applying ISENSE approach to a crowdtesting dataset show that with it significantly outperforms the two baselines, i.e. Rayleigh and Naïve method. Specifically, the median of predicted total bugs is nearly equal with the ground truth (i.e., $MRE < 0.06$) with a standard deviation of less than 10% during the latter half of the process. In addition, ISENSE can predict the required test cost within averagely 6% MRE for later stage of crowdtesting processes. Most significantly, the automation of task closing by ISENSE can make crowdtesting more cost-effective, i.e., **a median of 100% bugs can be detected with 30% cost reduction**. It also provides practical insights to help managers make cost-quality trade-off analysis on which task to close or when to close, based on two benchmark parameters and a set of decision heuristics.

The expected deliverables from the proposed research will enable time-sensitive, context-aware, value-driven decision making on resource allocation and utilization faced by testing managers in defense domains. It can be measured using the metrics such as testing efficiency and testing cost reduction through empirical data evaluation and pilot study.

1.5 RISKS AND MITIGATION PLANS

A major risk to the success of the proposed research is unable to identify testing data sources in the defense domain. The PI's previous work and the Incubator phase work mainly relied on software crowdtesting data collected from industry collaborators. As discussed in Section 2 and 3, most models and machine learning approaches constructed in MET are generic and can be easily extended to support a more sophisticated domain or special types of testing tasks. The project plans to further leverage existing dataset to emphasize on security testing and/or performance testing for mobile applications, and as the project proceeds, we will continue to seek out opportunities to identify other data sources to perform cross-dataset evaluation.

1.6 BUDGET

The estimated budget for Phase 2 is \$250K per year for two years. The primary cost is expected to be labor consisting of release time for the PI, one full-time graduate research assistants to conduct the research and develop models and approaches in MET, and part-time research assistant to help implement the tools in MET.

1.7 REFERENCES

1. W. E. Lewis, Software testing and continuous quality improvement. CRC press, 2016.
2. M. Garg, R. Lai, and S. J. Huang, "When to stop testing: a study from the perspective of software reliability models," IET software, vol. 5, no. 3, pp. 263–273, 2011.
3. SERC Workshop on Model-Based System Assurance: <https://sercuarc.org/event/serc-workshop-model-based-system-assurance/>
4. J. Wang, Y. Yang, R. Krishn, T. Menzies, Q. Wang, "iSENSE: Completion-Aware Crowdtesting Management," in Proc. of the Int'l Conf. on Software Engineering (ICSE'19), to appear.

5. Y. Yang, M. R. Karim, R. Saremi and G. Ruhe, "Who Should Take This Task? – Dynamic Decision Support for Crowd Workers", in: Proceedings of ESEM 2016.
6. S. H. Kan, Metrics and models in software quality engineering. Addison-Wesley Longman Publishing Co., Inc., 2002.
7. Jorgensen, Paul C. *Software testing: a craftsman's approach*. Auerbach Publications, 2013.
8. Schuler, David, and Andreas Zeller. "Assessing oracle quality with checked coverage." *2011 Fourth IEEE International Conference on Software Testing, Verification and Validation*. IEEE, 2011.
9. K. Mao, L. Capra, M. Harman, and Y. Jia, "A survey of the use of crowdsourcing in software engineering," *Journal of Systems and Software*, vol. 126, pp. 57–84, 2017.
10. X. Zhang, Y. Feng, D. Liu, Z. Chen, and B. Xu, "Research progress of crowdsourced software testing," *Journal of Soft-ware*, vol. 29(1), pp. 69–88, 2018.
11. <http://www.softwaretestinghelp.com/crowdsourced-testing-companies/>, 2018.
12. <https://dzone.com/articles/top-10-benefits-of-crowd-testing-1>, 2018.
13. <https://www.softwaretestinghelp.com/guide-to-crowdsourced-testing/>, 2018.
14. <https://dzone.com/articles/why-google-uses-crowd-testing>, 2018.
15. <https://www.applause.com/customers/>, 2018.
16. Karim R Lakhani, David A Garvin, and Eric Lonstein. Topcoder (a): Developing software throughcrowdsourcing. Harvard Business School General Management Unit Case, (610-032), 2010.
17. J. Wang, Y. Yang, S. Wang, Q. Wang, Context-aware In-process Crowdworke Recommendation. Submitted to International Conference on Software Engineering (ICSE) 2020, under review.
18. Junjie Wang, Song Wang, Jianfeng Chen, Tim Menzies, Qiang Cui, Miao Xie, and Qing Wang. Characterizing crowds to better optimize worker recommendation in crowdsourced testing. IEEE Transactions on Software Engineering, 2019.
19. Miao Xie, Qing Wang, Guowei Yang, and Mingshu Li. Cocoon: Crowdsourced testing quality maximization under context coverage constraint. In ISSRE'17, pages 316–327, 2017.
20. Qiang Cui, Junjie Wang, Guowei Yang, Miao Xie, Qing Wang, and Mingshu Li. Who should be selected to perform a task in crowdsourced testing? In COMPSAC'17, pages 75–84, 2017.
21. M. Zhou and A. Mockus. What make long term contributors: Willingness and opportunity in oss community. In ICSE'12, pages 518–528, 2012.
22. Zherui Cao, Yuan Tian, Tien-Duy B. Le, and David Lo. Rule-based specification mining leveraging learning to rank. *Autom. Softw. Eng.*, 25(3):501–530, 2018.
23. Haoran Niu, Iman Keivanloo, and Ying Zou. Learning to rank code examples for code search engines. *Empirical Software Engineering*, 22(1):259–291, 2017.
24. Qiao Huang, Emad Shihab, Xin Xia, David Lo, and Shanping Li. Identifying self-admitted technical debt in open source projects using text mining. *Empirical Software Engineering*, 23(1):418–451, 2018.
25. Junjie Wang, Qiang Cui, Qing Wang, and Song Wang. Towards effectively test report classification to assist crowdsourced testing. In ESEM'16, page 6, 2016.
26. Yang Feng, Zhenyu Chen, James A Jones, Chunrong Fang, and Baowen Xu. Test report prioritization to assist crowdsourced testing. In FSE'15, pages 225–236, 2015.
27. Yang Feng, James A Jones, Zhenyu Chen, and Chunrong Fang. Multi-objective test report prioritization using image understanding. In ASE'16, pages 202–213, 2016.
28. Qiang Wu, Christopher J. C. Burges, Krysta M. Svore, and Jianfeng Gao. Adapting boosting for information retrieval measures. *Information Retrieval*, 13(3):254–270, Jun 2010.
29. Chakkrit Tantithamthavorn, Shane McIntosh, Ahmed Hassan, and Kenichi Matsumoto. An empirical comparison of model validation techniques for defect prediction models. *TSE'16*, 43:1–18, 2016.
30. Yoshua Bengio, R'ejean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *The Journal of Machine Learning Research*, 3:1137–1155, 2003.
31. Xinli Yang, David Lo, Xin Xia, Lingfeng Bao, and Jianling Sun. Combining word embedding with information retrieval to recommend similar bug reports. In 2016 IEEE 27th International Symposium on Software Reliability Engineering (ISSRE), pages 127–137. IEEE, 2016.
32. Jayati Deshmukh, Sanjay Podder, Shubhashis Sengupta, Neville Dubash, et al. Towards accurate duplicate bug retrieval using deep learning techniques. In 2017 IEEE International conference on software maintenance and evolution (ICSME), pages 115–124. IEEE, 2017.

33. Alessandro Marchetto, Giuseppe Scanniello, and Angelo Susi. Combining code and requirements coverage with execution cost for test suite reduction. *IEEE Trans. Software Eng.*, 45(4):363–390, 2019.
34. Rohan Padhye, Caroline Lemieux, and Koushik Sen. JQF: coverage-guided property-based testing in java. In *Proceedings of the 28th ACM SIGSOFT International Symposium on Software Testing and Analysis, ISSTA 2019, Beijing, China, July 15-19, 2019.*, pages 398–401, 2019.
35. Michael Hilton, Jonathan Bell, and Darko Marinov. A large-scale study of test coverage evolution. In *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering, ASE 2018, Montpellier, France, September 3-7, 2018*, pages 53–63, 2018.
36. Thomas Oberlin and Rangasami L. Kashyap. 1973. Bayes Decision Rules Based on Objective Priors. *IEEE Trans. Systems, Man, and Cybernetics* 3, 4 (1973), 359–364.
37. Tranquillo, Joe. "The T-Shaped Engineer." *Journal of Engineering Education Transformations* 30.4 (2017): 12-24.

2. EXPLAINING MODEL COMPOSABILITY USING CAUSAL GRAPHS— DR. MICHAEL PENNOCK, STEVENS INSTITUTE OF TECHNOLOGY

2.1 PROBLEM STATEMENT, MOTIVATIONS, AND OBJECTIVES

Computational modeling has become central to engineering design, and it is particularly important to the US Department of Defense (DoD). Over the past several decades, DoD has invested billions in developing computational models to support its mission, but these models are typically purpose built and can rarely be adapted to new applications. A prior effort to reuse and integrate existing DoD simulations ended when it was determined to be infeasible [1]. The problem is becoming particularly acute as threat scenarios are evolving faster than new models can be developed or adapted to explore new strategies or enable the development of new systems to address them. DoD requires a capability to rapidly develop and validate computational models would enable to DoD to computationally analyze and respond to rapidly evolving threat scenarios and accelerate DoD’s system design and test cycle. One approach to accomplish this would be to construct libraries of reusable simulation modules that could be selectively combined to create new models as needed with minimal adaptation. Unfortunately, a 2017 National Science Foundation (NSF) workshop report on modeling and simulation noted that model reuse is “peculiarly fragile” [2].

While experienced system modelers may have intuitive explanations for why this is so difficult, an explanatory theory is required to: 1) determine when it is and is not appropriate to compose diverse sets of models, 2) develop methods, processes, and tools to enable composition when it is appropriate. The long-term goal of this line of research is to determine if adapting an approach from computer science, causal graphs, can explain why combining diverse computational engineering models works in some circumstances but not others. The specific objective of this incubator project is to investigate the feasibility of this approach by testing whether causal graph

analysis methods can detect composability issues experienced during a previously completed simulation study in the healthcare domain.

What was found during the incubator project is that the application of causal graph methods is feasible, but there are a number of modifications that would need to be made to how the graphs are analyzed. More specifically, the frontdoor adjustment set shows promise as a means to identify candidate modules, but the algorithm for identifying the adjustment set would need to be evolved to handle multiple inputs, outputs, and layers of abstraction. Furthermore, several potential use cases were identified including representing a domain of interest as a causal graph and identifying likely reusable modules, checking the viability of existing modules against the domain graph, and developing targeted tests to evaluating black-box simulation modules for reusability. These would all need to be evaluated during the next phase of the research.

2.2 CURRENT PRACTICE AND ITS LIMITATIONS

Model reuse and composition has been accomplished in domain specific modeling frameworks and multi-scale modeling. Domain specific frameworks involve decomposing and standardizing modeling components to meet the needs of a particular domain such as infrastructure modeling [3], logistics systems [4], healthcare [5], and space science missions [6]. Multi-scale modeling involves connecting models from different scales of resolution via scale bridging [7, 8]. Scale bridging is accomplished via “handshake” algorithms that are usually domain and/or application specific and may require experimental data to tune [9, 7, 8]. In some cases, this is done for computational advantage, and in others this is done because there is no known single model that can capture the phenomena. Example applications include cancer biology [10], in-stent restenosis [11], chemical reactors [12], and chemical process equipment [13]. The limitations of these approaches are that they are largely idiosyncratic and require a great deal of trial and error to implement.

While the problem of determining model consistency and composability may seem conceptually simple, in practice, it has proven quite challenging. Taylor, et al. assert that “Composability is still our biggest simulation challenge.” [14, p. 652]. Particularly problematic is the fact that the combination of two valid models is not necessarily valid [15]. The ability to exchange data between models is not sufficient to achieve a valid combination of models. It turns out that model interoperability itself is a multi-layered problem. The Levels of Conceptual Interoperability Model (LCIM) describes several abstract layers of interoperability such as syntactic, semantic, pragmatic, dynamic, and conceptual [16, 17]. These are useful as broad guides, but LCIM does not provide specific rules for model composition. So while there are existing standards and theories for coordinated model execution and data exchange [18, 19] (i.e. dynamic and syntactic), other layers such as semantic and conceptual remain problematic. Highlighting a similar problem in software reuse, Garlan et al. attribute these sorts of interoperability issues to inconsistent assumptions about operating environments embedded in the design of the modules, a phenomenon they call “architectural mismatch” [20, 21].

The previously mentioned NSF M&S workshop identified two findings that summarize the

research needed to address these challenges [2]:

- “Finding D.1 Advancements in the theory of reuse are needed to provide a firm theoretical foundation for producing robust and reliable reuse practices.”
- “Finding D.2 Guides of good practices on reuse of simulation solutions, data, and knowledge discovery can in particular support the workforce.”

These findings provide the motivation behind the research described here.

2.3 RESEARCH APPROACH

2.3.1 Applying Causal Graphs to Computational Models

The focus of this research is to understand aspects of semantic composability. As indicated above, the challenge of model composability is not addressed by ensuring consistent interfaces among component models. That is necessary, but not sufficient. Rather the issue is that when models are assembled, they do not generate valid results that are consistent with the system being modeled. To identify potential sources of inconsistency during model composition, this research employs a technique called a causal graph to describe and analyze the reference system to be modeled.

Causal graphs are an evolution of Bayesian networks and techniques have been developed to analyze these graphs to facilitate the extraction of causal effects from observational and experimental data sets (Figure 1). When a causal effect can be isolated, it is called a mechanism, and it can be reused to understand the impact of interventions. Causal diagrams have already shown success in fusing heterogeneous statistical models and data sets [22]. The rationale behind applying causal graphs to simulation models is based on an insight from algorithmic information theory that models can be viewed as compressed data [23]. From this perspective, it stands to reason that an approach that can be used to assess composability of data sets can also be used to assess the composability of computational models. In other words, if it were not possible to isolate a behavior interest from a particular data set describing the system, then how would it be possible using the compressed version of that data set, the computational model?

One of the advantages of the causal graph is that we can use a condition called d-separation to identify conditional independence. When a set of graph nodes X is d-separated from a set of graph nodes Y given another set of nodes Z , it means that we can analyze X and Y independently as long as we know the values of Z . In other words, they are causally isolated. The idea of d-separation can be extended to yield the backdoor criterion. The backdoor criterion determines which sets of nodes must be controlled to identify a causal effect. This set is called a backdoor adjustment set. Repeated applications of the backdoor criterion can be used to determine when a causal effect can be isolated as a mechanism. The result is called the frontdoor criterion. Backdoor and frontdoor adjustment sets can be identified by analyzing the structure of the graph alone. There is an equivalent set of operations that can be performed on stochastic equations associated with the graph. These operations are collectively known as the do-calculus.

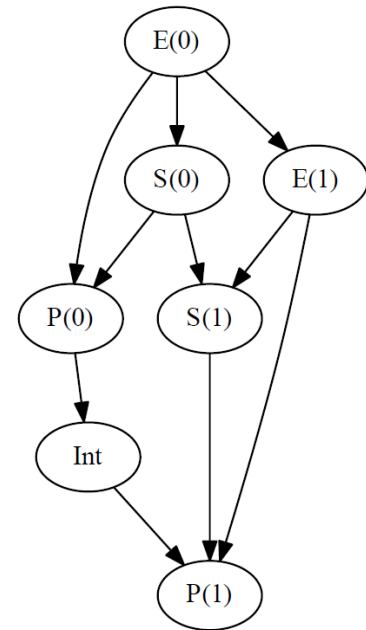


Figure 1 - An example of a causal graph

Consider Figure 2. In this simple example we are interested in understanding which types of patients, “P,” would benefit from an intervention “Int.” The issue is that the ultimate impact of the intervention is confounded by the economic “E” and social “S” environments of the patient. To determine which patients would benefit, we choose patients in a particular state at time 0. In the terminology of causal graphs, we apply the $do()$ operator to the node “P(0)” indicated by the green box. We then want to observe the effects on the states of those patients at time 1, indicated by the node “P(1).” The problem we have is that we do not know whether any measured impact is a result of the patient selection and intervention or the environmental and social conditions outside of our control.

If we apply the backdoor criterion to the graph, we find that the set of nodes “E(0)” and “S(0)” are a backdoor adjustment set shown as the red box. This means that if we could control for the values of “E(0)” and “S(0)” in the observational data set, we could isolate the impact of applying the intervention to different types of patients. While important from a statistical standpoint, the backdoor criterion does not identify modules. However, if we apply frontdoor criterion to the graph, we find the node “Int” is a frontdoor adjustment set shown as the blue box. This means that “Int” is a mechanism, and we can apply it independently of the social and economic context to understand impact of choosing different types of patients. This is not to suggest that economic and social contexts are not important, but it does mean that we could reuse “Int” with different models of social and economic context without having to change it. Thus, it is modular. For a more in-depth discussion of casual graphs and the techniques used to analyze them, see [24].

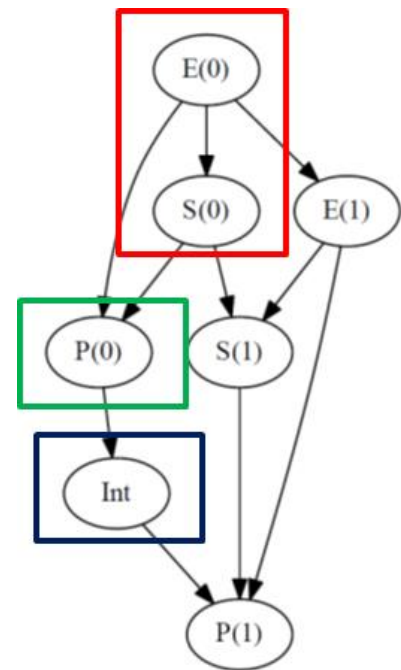


Figure 2 - Examples of the frontdoor and backdoor adjustment sets

While the fact that “Int” is isolated in this graph may seem obvious in this simple example, it becomes less obvious in larger, more complicated graphs. Also, seemingly small changes in the structure of the graph can alter whether “Int” is a frontdoor adjustment set. For example, if it is determined that the social context affects the intervention, which is true for some interventions, then we need to add an arc from “S(0)” to “Int,” and it is no longer a frontdoor adjustment set. Thus, it cannot be modeled independently of social context.

To apply these techniques to computational models, it is important to note that they only require that the nodes represent functions. In fact, these techniques have already been applied to a type of linear equation based model called a structural causal model, which is used in economics [24]. Though, the equations do not need to be linear. The graph nodes can represent any functional form. Since computational models are compositions of functions, the application follows naturally. The approach taken in this line of research is to describe the system we wish to model computationally using a causal graph. Then, we can apply the techniques described above to determine whether or not we can model the system using a composed set of computational modules.

It is important to note that the graph itself is a form of modularization because it presumes that the real-world phenomena of interest can be represented by a discrete set of interconnected functions. Thus, if one could construct an accurate set of computational functions for each node in the graph, then one could compose the model of the system. There are two issues here. First, for a very complex system with a complex graph, we would want to reuse entire chunks of the model rather than each atomic function by itself. In the limit, only reusing the atomic functions would not gain much beyond building a new model from scratch. Second, it may not be possible to computationally model every node in the graph as the functional structure of some nodes may

be unknown. In the prior example, we might not have a good model of the social context of the patient or its impact on the patient's health. If we were to construct a simulation model of the intervention that ignored social context and then validated it against available data, we would be locking in any idiosyncratic features of social context latent in the available data. This means that we could not reliably reuse the model in a different context. This is where the frontdoor criterion is critical. If the intervention model is a frontdoor adjustment set, then we can reuse the model because it is causally isolated from the social context. Following this line of reasoning, if we can identify frontdoor adjustment sets in the causal graph of a system, these sets would be candidate modules that could be described with a computational model and reused.

Beyond assisting in the decomposition of a computational model into reusable modules, causal graphs of the reference system have the potential to determine which if any combination of existing modules can be composed to represent it accurately, analyze how a shift in context or behavior might affect existing simulations, and develop targeted experiments to make inferences about the causal structure of a black-box model.

2.3.2 Research Approach for the Incubator Project

As noted previously, the specific objective of this incubator project is to assess the feasibility of applying causal graph techniques to understand model composability. Assuming feasibility is established, the output would be a plan to test the approach more rigorously. The research approach for the incubator is to apply causal graph techniques to a real modeling problem as a small-scale test. The test uses an existing simulation developed for a previous research project in the healthcare domain [25, 26]. This simulation models the adoption of a healthcare intervention called the Transitional Care Model (TCM). The reason for using an existing, completed simulation project is three-fold. First, necessary artifacts describing the reference system have already been created, though not in the form of causal graphs. Second, the outcome of the study and simulations are known. That enables a comparison of the design implications resulting from the causal graph analysis with the design choices actually made for the real simulation along with all associated difficulties encountered. If there is consistency among the predictions of the causal graph analysis and real-world outcomes of the simulation study, it suggests that the causal graph approach is at least feasible and worthy of further study. Third, the system is complex enough to be interesting. Application of the causal graph methods to simple graphs yield fairly obvious outcomes that do not necessarily scale to more realistic system graphs.

The specific steps of the incubator project are:

1. Construct an ontology and process model for the problem domain of the TCM investigation.
2. Convert the ontology and process model into a causal graph.
3. Test the application of d-separation, frontdoor adjustment, and backdoor adjustment to identify candidate graph decompositions.
4. Analyze how various applications of the "do()" operator alter the decompositions.

5. Compare the candidate decompositions against the known outcomes of the prior healthcare simulation study.
6. Based on the findings, develop a plan to test the approach on a large-scale systems modeling problem.

The ontology and process models were developed using SysML in Sparx Enterprise Architect. The causal graphs were built and analyzed using the open source Python package CausalGraphicalModels (<https://pypi.org/project/causalgraphicalmodels/>).

2.4 RESULTS OF THE INCUBATOR STUDY

2.4.1 Constructing and Analyzing the Causal Graphs

During the prior TCM simulation study, information about the structure of the reference system was collected via structured interviews with key stakeholders. Interview results were converted into a set of influence diagrams to capture key decisions, variables, outcomes, and the relationships among them. Converting this material into a causal graph required a more rigorous and precise description of the system than was present in the existing influence diagrams. To address this, the information from the influence diagrams was distilled and captured using SysML block diagrams and activity diagrams. The block diagrams were used to construct an ontology of the system to describe the key entities and relationships among them. An example is shown in Figure 3. Activity diagrams with object flows were used to describe the processes within the healthcare system being modeled. An example is shown in Figure 4. The objects that flow along the arcs in the activity diagram are from the block diagrams. The activities in the activity diagrams naturally align with functions, which facilitates the conversion to causal diagrams.

To convert the activity diagrams to causal graphs, each low-level activity was assigned a function node on the graph. Furthermore, each decision node in an activity diagram was also assigned a function node. The entities that moved along the arcs among the activities were viewed as state variables that are modified by the functions. Since a causal graph must be acyclic, cycles in the activity diagrams were removed by unfolding the graph and assigning a time index to each node. For example, the function to update patient state, P , was converted to graph nodes $P(0)$, $P(1)$, $P(2)$, etc. in order to remove the cycles associated with the activity “Update Patient Condition.”

The end result of the conversion process was a fairly large causal graph. An excerpt from the base-case graph is shown in Figure 5.

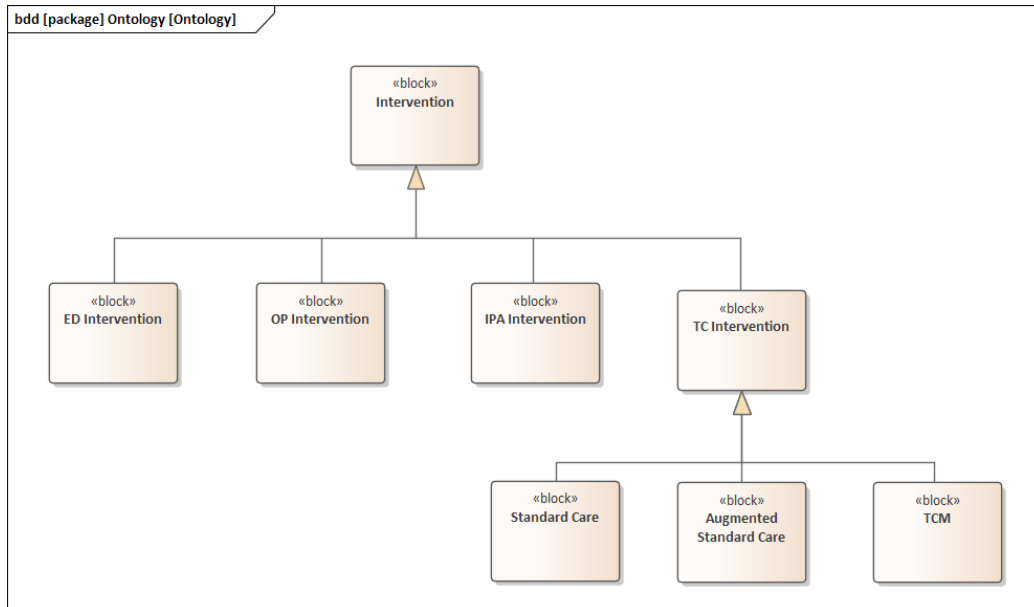


Figure 3 - Ontology for Interventions

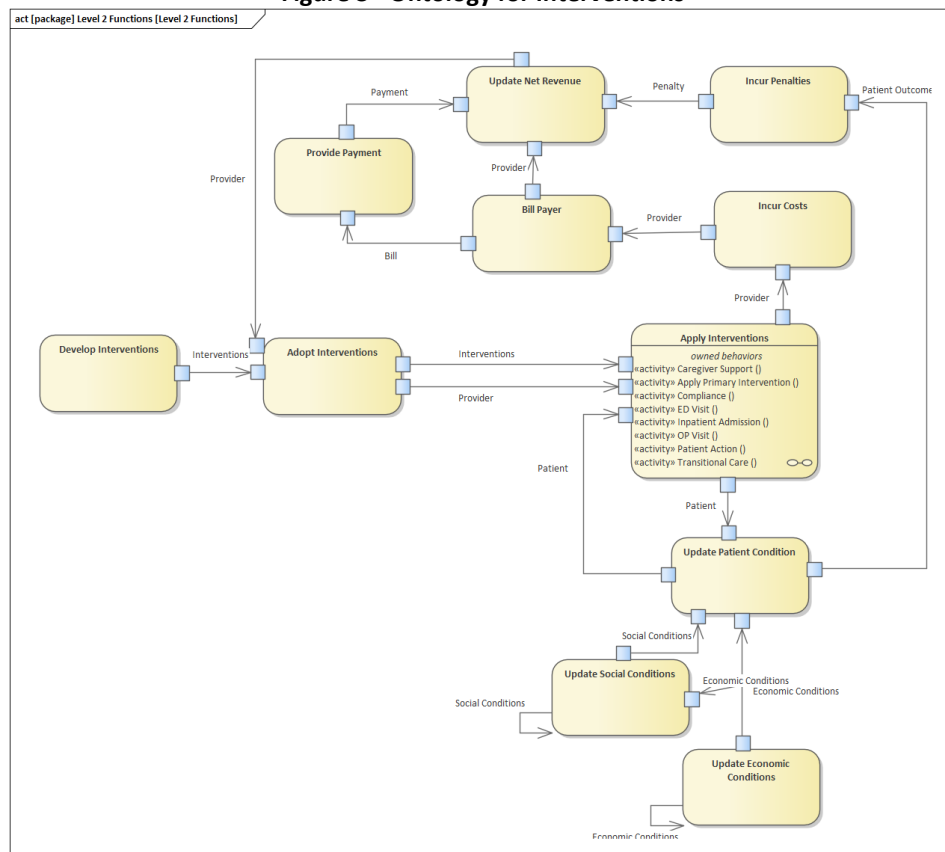


Figure 4 - Example process model

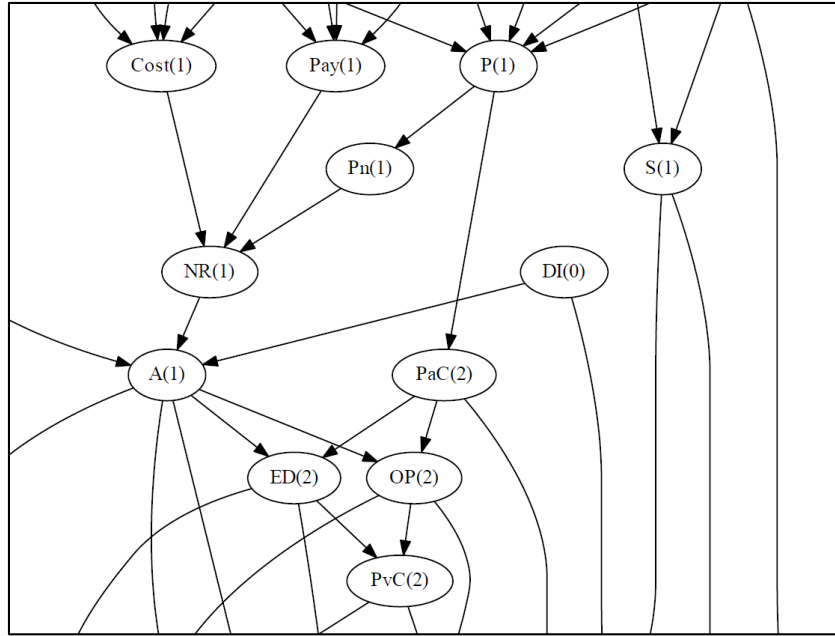


Figure 5 - Excerpt from the causal graph for the reference system in which TCM was evaluated

Following the creation of the causal graph, a number of tests were run to understand how d-separation, the backdoor criterion, and frontdoor criterion were manifested in the causal graph. It is important to note that one of the key ideas of causal graph analysis is that the scenario of interest can actually alter the structure of the graph. For instance, if we want to intervene on a particular function, we are actually altering the function and removing input links for that function. For example, if we want to understand the impact of adopting a particular healthcare intervention, that is equivalent to replacing the intervention choice function in the graph with a pre-determined choice. This necessarily removes incoming arcs of information that would have otherwise influenced the doctor's choice of intervention as these are now irrelevant. Thus, the structure of the graph is now different. For this reason, backdoor and frontdoor adjustment sets can only be identified once the decision nodes and target nodes for a scenario of interest are identified.

In the case of the TCM study, there were two broad questions of interest. First, what is impact of adopting TCM for a given hospital and specific set of patients in terms of patient outcomes and financial impacts? Second, what factors affect the widespread adoption of TCM across the US healthcare system? Each of these have different decision and outcome nodes of interest.

For the first question, we set $P(1)$ to a decision node to indicate that we are interested in patients with specific conditions. We also set all of the adoption nodes $A(i)$ to decision nodes as we want to see what happens when we force the use of TCM on the specified patient group. The response variables of interest were designated as the patient outcome $P(2)$ and the net revenue of the provider $NR(2)$. With the source and target variables set, it was possible to find all backdoor and frontdoor adjustment sets.

During the tests, three difficulties were encountered. First, the algorithms in the causalgraphicalmodels package could not handle multiple source and target nodes simultaneously. Second, the algorithms used to find all frontdoor adjustment sets and all backdoor adjustments sets failed to terminate. These algorithms were previously tested successfully on smaller graphs, but there seemed to be computational issues on the larger TCM test graph. Fortunately, the algorithms that checked proposed backdoor and frontdoor adjustment sets still worked correctly on the large TCM graph. Third, the frontdoor adjustment algorithm could not detect cases where a subgraph of nodes would be a frontdoor adjustment set if it were replaced with a single node. These three difficulties should not be construed as problems with the causalgraphicalmodels package as this study was applying the algorithms outside of their intended use cases. However, with some adjustments, the tests were able to proceed.

To adjust for the multiple input nodes, the graphs were manually updated to for all but one of the input nodes to create a graph structure equivalent to the case where the other input nodes had been fixed. Checking multiple output nodes was simpler to address as it is equivalent to checking them one at a time. To identify adjustment sets, these were hypothesized and then checked using the algorithms in the causalgraphicalmodels package. While this did not constitute an exhaustive search, it was sufficient to establish feasibility. Finally, to detect frontdoor adjustment sets that consist of an entire subgraph, these were hypothesized and manually replaced in the graph with a single node representing the entire subgraph. The frontdoor criterion was then checked against this modified graph.

2.4.2 Results of the Main Test Cases

Searching for the frontdoor adjustment sets in the main test cases provided the most interesting and relevant results. First, the subgraph associated with applying interventions was found to be a valid frontdoor adjustment set when setting the patient state and intervention option (Figure 6a). Another subgraph related to updating the hospital's finances was also found to meet the frontdoor criterion assuming that we set the interventions applied and the performance penalties imposed by the US Government (Figure 6b).

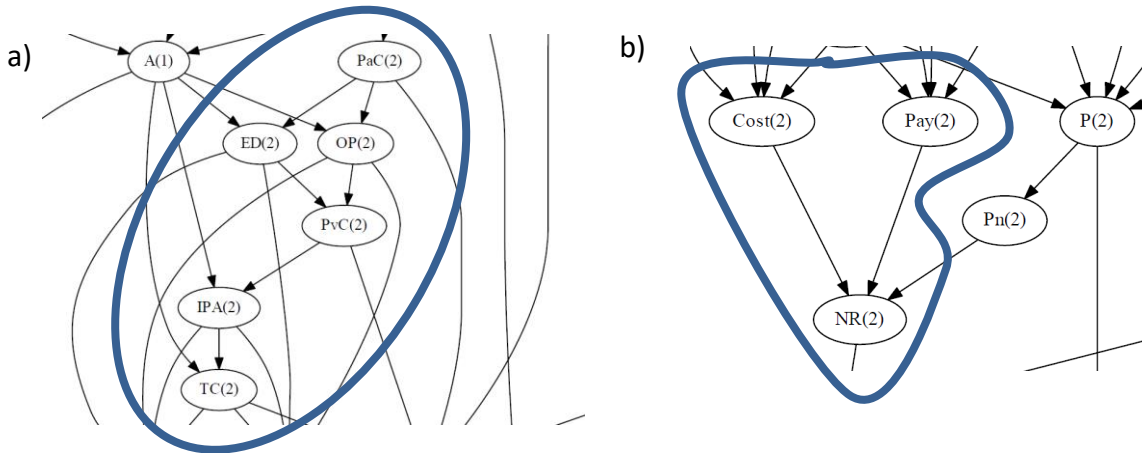


Figure 6 – a) Frontdoor adjustment set for applying interventions assuming A(1) is fixed, b) Frontdoor adjustment set for updating the hospital's financials assuming Pn(2) is fixed

If we assume that the frontdoor adjustment set is a good candidate for a module, then each of these subgraphs could be converted to a module and reused. Using these two modules would let us model the financial and patient impacts of various healthcare interventions beyond just TCM. But again, this depends on the scenarios of interest and which nodes are controlled as decision variables.

If we consider the second of question of interest, what factors affect the widespread adoption of TCM by hospitals, the graph we developed for the first question is essentially repeated, one for each hospital, with some interconnections among them. Now the decision variable is the deliberate introduction of TCM at the D(i) nodes of the graph. The intervention adoption nodes A(i) are now central to the model. Now that these are no longer fixed, the frontdoor criterion is violated for the intervention module identified in the first case.

Interestingly, this result bears some similarity to the outcomes experienced during the TCM simulation study. Initially, the plan was to develop single simulation to address both questions of interest. Very early in the simulation development effort it was determined that this was not feasible. The differences in perspective for each question meant that we were unable to simply reuse the intervention code to answer both questions. Ultimately, we ended up developing two different simulations with different implementations of how the interventions were handled. This was a direct result of the fact that to answer the widespread adoption question, we needed to have the ability handle multiple different interventions selected and applied simultaneously.

It should be noted that there was some reuse among the simulations. The model of TCM's impact on individual patients was effectively the same in both simulations as they were both drawn from the same TCM randomized control trial results. However, this reuse was at a much lower level than the large modules identified above. From the viewpoint of the causal graph, this would be equivalent to decomposing the TC(i) node into a subgraph where the specific impact of TCM on a patient could be isolated as a mechanism that satisfies the frontdoor criterion.

In addition to the base case tests described above, the casual graphs could also be used to illustrate a point of contention that arose during the TCM simulation study. During reviews of the simulations with healthcare providers, one of the providers asserted that, based on their experience, the impacts of TCM are not independent of the social and economic and economic context of the patient. In terms of the causal graph, this assertion would be equivalent to drawing arcs from the S(i) and E(i) nodes to the intervention nodes. Figure 7 illustrates this for the S(1) node in the causal graph where the intervention subgraph has been rolled up to a single reusable module, Int(2). This connection now violates the frontdoor criterion and Int(i) is no longer a mechanism to update patient states.

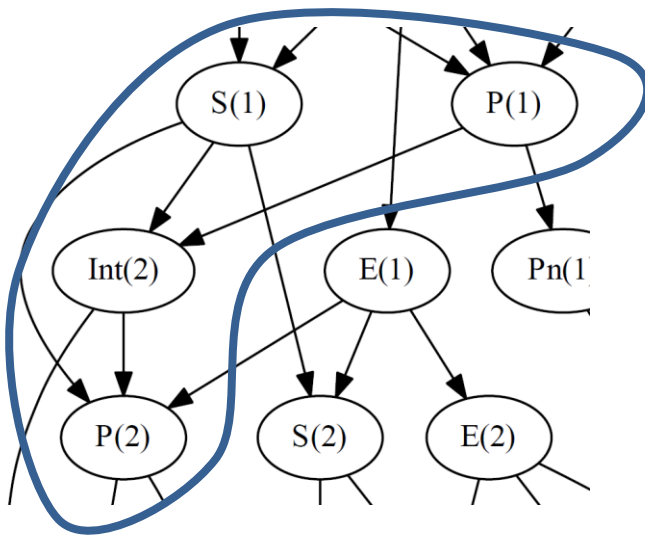


Figure 7 – Arc added from social context S(1) to the rolled up intervention node Int(2) violates the frontdoor criterion for Int(2)

If this this assertion is correct, then the ability to apply our simulation of the impacts of TCM on hospitals would be limited to those that had patients with social and economic characteristics similar to where the RCTs were conducted. This would adversely impact the reusability of our simulation in other contexts or for other interventions. The counterargument made by the researchers that developed TCM is that they specifically tested for the impact of social, demographic, and economic factors and did not find statistically significant differences.

Regardless, this analysis highlights the ability of a causal graph to immediately highlight a potential issue with reusing a simulation module. If we compare the causal graph of this scenario with associated the SysML activity diagram, we see that the problem is not obvious in the SysML diagram. Figure 8 shows an excerpt from the activity diagram highlighting the application of the interventions. a) shows the original model and b) adds a flow from “social conditions” to “apply interventions.” The structure of the activity diagram implies that “apply interventions” is a functional module since it contains a number of sub-activities and accounts for all inputs and outputs. The issue is how “social conditions” connects to both “apply

interventions” and “update patient condition.” This creates a confounding effect that is not obvious from an examination of “apply interventions” alone. It may still be possible to control for the social conditions as it is part of a backdoor adjustment set in the updated model, but that would require creating an accurate model of social conditions, a challenging proposition at least for the case of TCM.

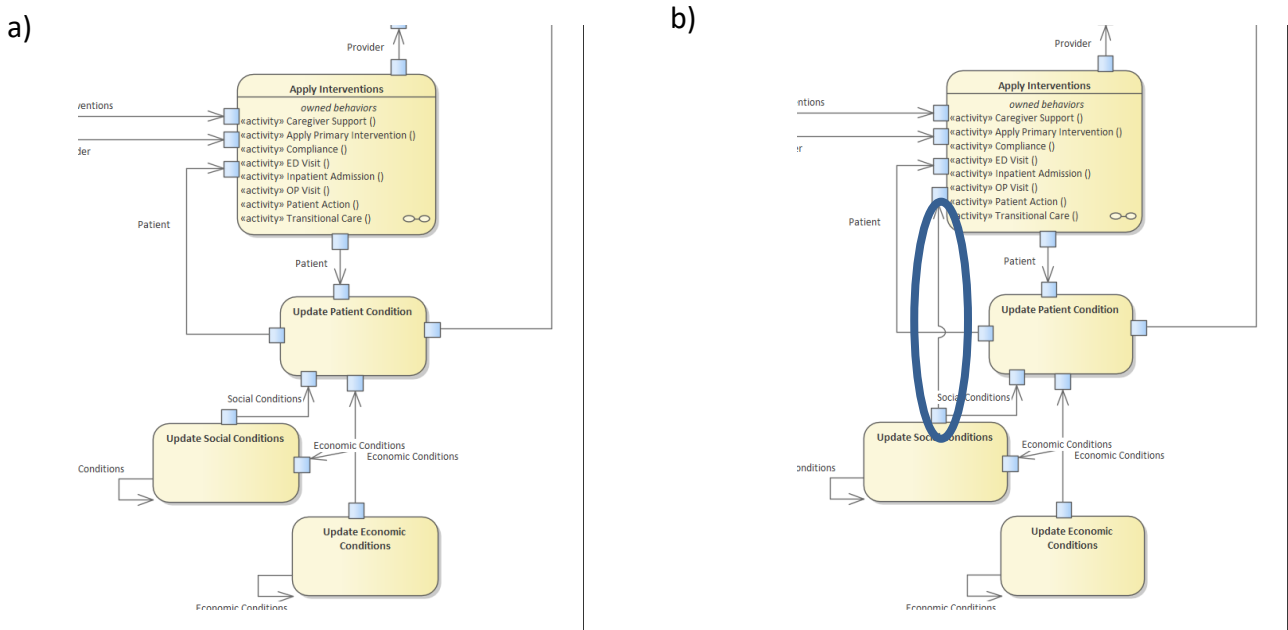


Figure 8 – a) The original process model for applying interventions. b) The process model with a new connection showing social conditions affecting the interventions

2.4.3 Additional Excursions

Based on observations made during the main test cases, a number of small-scale excursions were conducted to explore interesting behaviors. Findings of interest include:

- There are many combinations of nodes that are d-separated in a large graph. Not all are equally useful to identifying potential modules. The backdoor and frontdoor criterion seem to produce more useful sets of nodes.
- Searching for all possible backdoor adjustment sets can yield a large number of sets. Many are overlapping and not all are useful. It would be useful to develop criteria to narrow the set down to those most relevant to modularity
- Backdoor adjustment sets seem to be good candidates for what Tolk called “ghosted objects” [27]. Ghosted objects are objects that are relevant to multiple models in a composite simulation but should only be modeled once to maintain consistency.

2.4.4 Discussion

Overall, the results of the incubator project indicate the application of causal graphs to understand model composability is feasible. While the evidence from a feasibility study is not sufficient to definitively establish the utility of the approach, it is enough to show that it merits further investigation and provides insight as to how to conduct more in-depth investigations. Perhaps the most interesting result of this study is a qualitative one. During the course of the study, it was found that even small, reasonable changes to the structure of the causal graph could impact whether a group of nodes met the frontdoor criterion. Similarly, changing the scenario of interest could also change the frontdoor adjustment sets. If we consider a mechanism a necessary condition for a reusable module, then the analysis of causal graphs yields a similar fragility to what we see when we attempt reuse and compose computational models in real life. The slightest change in context could invalidate the semantic consistency of the module.

This outcome also suggests what types of systems would be easier to simulate using reusable/reconfigurable modules. Systems of this type would have causal graphs that consist of relatively isolated sequences of mechanisms with little crosslinking. Communication would occur consistently through intermediate nodes between the mechanisms. One could hypothesize that that this structure is more likely to occur in low-level physics-based systems. Socio-technical systems, on the other hand, are unlikely to exhibit this structure, and historically, we have found these types of systems very difficult to model. Still, even in socio-technical systems there may be certain layers of abstraction where there is a sufficient amount of stability to enable reusable models.

To manage this fragility of computational model reusability, the results of the study suggest a few potential use cases for causal graphs to facilitate the reusability of computational modules. First, a comprehensive causal graph could be developed for a domain of interest. This graph could be scanned for frontdoor adjustment sets over a range of different scenarios. Frontdoor adjustment sets that remain stable across many scenarios are good candidates for reusable simulation modules that could be archived in a library for future reuse.

Second, if the modules in a reusable simulation library have well documented causal graphs, these could be used to determine what combination of modules could be composed to model a target system. To do this, a causal graph would be developed that describes the target system. This graph could then be analyzed to determine if it could be reconstructed using a combination of the graphs describing the available modules. In the event of a close, but imperfect match, the graph should reveal what modifications may be necessary.

Third, for existing legacy simulations where the internal assumptions are not fully known, it may be possible to infer the causal graph, or at least the portions relevant for reuse, from targeted tests. In a sense, this is similar to the type of inferences made when randomized control trials are performed to understand human biology. A set of hypothesized causal graphs could be created based on potential alternative implementations of the simulation. A set of tests could then be devised that would determine which, if any, of the hypothesized causal

graphs accurately describe the legacy simulation. This approach has the potential to be more effective than exhaustive integration testing.

Finally, it is also important to note that nothing in the work described is strictly limited to the composition of simulation modules. All of the same principles should apply to the composition of existing systems into systems of systems or cyber-physical systems.

2.5 PROPOSED WORK FOR THE NEXT PHASE

In the next phase, it is necessary to apply the methods to an even larger scale system that will yield more than two modules. This will be necessary for any realistic assessment of the validity of the method. This system should be an engineered or physical system as this is less likely to have the chaotic structure of a sociotechnical system. The goal would be to construct a causal graph describing the target system and then identify potential modules. These modules would then be implemented as a limited simulation to test the ability to mix and match the modules and still yield valid results.

To enable this effort, a few preliminary tasks would need to be conducted:

1. The algorithms used to identify frontdoor and backdoor adjustment sets would need to be evolved to handle multiple input and output nodes. Also, they would need to be made more robust for large graphs. This may be achievable by modifying the algorithm to selectively abstract out portions of the graph and only focus on local subgraphs.
2. The conversion of SysML diagrams to casual graphs needs to be automated to handle a large system. As an alternative, Object-Process Methodology (OPM) graphs could be used as these may be easier to convert to causal graphs.
3. The analysis methods need to be modified to search across multiple levels of abstraction. It is likely that certain levels of abstraction are more amenable to modularization than others. OPM may be more useful than SysML in this regard.
4. Develop a method to quickly analyze the impact of perturbations in graph structure on the modular structure through targeted analysis of the causal graph

After the completion of these preliminary tasks, the application of the methods to the test system could begin. The steps are as follows:

1. Review and analyze system documentation
2. Construct a comprehensive model of the system using SysML or OPM
3. Convert the system model into a causal graph
4. Use the algorithms developed in the preliminary tasks to identify candidate modules
5. Choose a reasonable subset of the potential modules and implement them as simulations
6. Run experiments where the implemented modules are mixed and matched to simulate different system scenarios
7. Check the composed models for validity

2.6 REFERENCES

- [1] F. E. Mullen, "Dynamic multilevel modeling framework phase 1 - feasibility," US Department of Defense, Modeling and Simulation Coordination Office, Washington, DC, 2013.
- [2] R. Fujimoto, C. Bock, W. Chen, E. Page and J. H. Panchal, Eds., Research challenges in modeling and simulation for engineering complex systems, Cham: Springer International Publishing, 2017.
- [3] P. T. Grogan and O. L. de Weck, "The ISOs modeling framework for infrastructure systems simulation," *IEEE Systems Journal*, vol. 9, no. 4, pp. 1139-1150, 2015.
- [4] T. Sprock and L. F. McGinnis, "Simulation model generation of discrete event logistics systems (DELS) using software design patterns," in *Proceedings of the 2014 Winter Simulation Conference*, 2014.
- [5] B. P. Zeigler, "Discrete event system specification framework for self-improving healthcare service systems," *IEEE Systems Journal*, vol. 12, no. 1, pp. 196-207, 2018.
- [6] T. J. Bayer, M. Bennett, C. L. Delp, D. Dvorak, J. S. Jenkins and S. Mandutianu, "Update-concept of operations for Integrated Model-Centric Engineering at JPL," in *2011 IEEE Aerospace Conference*, 2011.
- [7] B. Chopard, J. Borgdorff and A. G. Hoekstra, "A framework for multi-scale modelling," *Philosophical Transactions of the Royal Society A*, vol. 372, no. 2021, 2014.
- [8] A. Hoekstra, B. Chopard and P. Coveney, "Multiscale modelling and simulation: a position paper," *Philosophical Transactions of the Royal Society A*, vol. 372, no. 2021, 2014.
- [9] E. Winsberg, Science in the age of computer simulation, University of Chicago Press, 2010.
- [10] Y. Liu, J. Purvis, A. Shih, J. Weinstein, N. Agrawal and R. Radhakrishnan, "A multiscale computational approach to dissect early events in the Erb family receptor mediated activation, differential signaling, and relevance to oncogenic transformations," *Annals of Biomedical Engineering*, vol. 35, no. 6, pp. 1012-1025, 2007.
- [11] A. Caiazzo, D. Evans, J. L. Falcone, J. Hegewald, E. Lorenz, B. Stahl, D. Wang, J. Bernsdorf, B. Chopard, J. Gunn and R. Hose, "A complex automata approach for in-stent restenosis: two-dimensional multiscale modelling and simulations," *Journal of Computational Science*, vol. 2, no. 1, pp. 9-17, 2011.
- [12] D. G. Vlachos, "Multiscale integration hybrid algorithms for homogeneous–heterogeneous reactors," *AIChE Journal*, vol. 43, no. 11, pp. 3031-3041, 1997.
- [13] A. Yang and W. Marquardt, "An ontological conceptualization of multiscale models," *Computers & Chemical Engineering*, vol. 33, no. 4, pp. 822-837, 2009.

- [14] S. J. Taylor, A. Khan, K. L. Morse, A. Tolk, L. Yilmaz, J. Zander and P. J. Mosterman, "Grand challenges for modeling and simulation: simulation everywhere—from cyberinfrastructure to clouds to citizens," *Simulation*, vol. 91, no. 7, pp. 648-665, 2015.
- [15] E. W. Weisel, R. R. Mielke and M. D. Petty, "Validity of Models and Classes of Models in Semantic Composability," in *Proceedings of the Fall 2003 Simulation Interoperability Workshop*, Orlando, FL, 2003.
- [16] W. Wang, A. Tolk and W. Wang, "The levels of conceptual interoperability model: applying systems engineering principles to M&S," in *Proceedings of the 2009 Spring Simulation Multiconference*, 2009.
- [17] A. Tolk and J. A. Muguira, "The levels of conceptual interoperability model," in *Proceedings of the 2003 fall simulation interoperability workshop*, 2003.
- [18] IEEE, "IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)-- Framework and Rules," IEEE, 2010.
- [19] B. P. Zeigler, H. Praehofer and T. G. Kim, *The Theory of Modeling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems*, 2 ed., Academic Press, 2000.
- [20] D. Garlan, R. Allen and J. Ockerbloom, "Architectural mismatch: Why reuse is still so hard," *IEEE Software*, vol. 26, no. 4, pp. 66-69, 2009.
- [21] D. Garlan, R. Allen and J. Ockerbloom, "Architectural mismatch or why it's hard to build systems out of existing parts," in *Proceedings of the 17th international conference on Software engineering*, 1995.
- [22] E. Bareinboim and J. Pearl, "Causal inference and the data-fusion problem," *Proceedings of the National Academy of Sciences*, vol. 113, no. 27, pp. 7345-7352, 2016.
- [23] G. Chaitin, *Meta Math!: The Quest for Omega*, Vintage, 2006.
- [24] J. Pearl, *Causality: models, reasoning and inference*, Cambridge: MIT Press, 2000.
- [25] M. J. Pennock, Z. Yu, Hirschman, K. B, K. Pepe, M. V. Pauly, M. D. Naylor and W. B. Rouse, "Developing a policy flight simulator to facilitate the adoption of an evidence-based intervention," *IEEE journal of translational engineering in health and medicine*, vol. 6, pp. 1-12, 2018.
- [26] W. B. Rouse, M. D. Naylor, Z. Yu, M. J. Pennock, K. B. Hirschman, M. V. Pauly and K. M. Pepe, "Policy Flight Simulators: Accelerating Decisions to Adopt Evidence-Based Health Interventions," *Journal of Healthcare Management*, vol. 64, no. 4, pp. 231-241, 2019.
- [27] A. Tolk, "The elusiveness of simulation interoperability: what is different from other interoperability domains?," in *Proceedings of the 2018 Winter Simulation Conference*, 2018.

3. TEST, EVALUATION, VERIFICATION AND VALIDATION (V&V AND T&E) COMPETENCY FRAMEWORK – DR. LAURA FREEMAN, VIRGINIA TECH

3.1 OVERVIEW AND MOTIVATION

3.1.1 Overview

The discipline of systems engineering currently lacks a competency framework for verifying and validating complex systems through a test and evaluation process. The overarching goal of this research program is to develop a model of the key competencies required for the DoD acquisition workforce to support Verification & Validation (V&V) through Test & Evaluation (T&E). In doing so, the research will set a new paradigm for transforming engineering education and professional training in the areas of V&V and T&E. Specifically, the objectives of this incubator project were to develop an initial structure of a V&V and T&E competency framework, establish the research methodology that we will employ to create and validate the framework, and identify the information sources that we will use to perform the study.

This paper outlines the results of the incubator project, which reviewed existing academic disciplines, existing competency frameworks, and current organizational structures. We provide an initial framework based on this review as well as a validation strategy moving forward. The paper outlines how a larger research program has the potential to reinvent V&V and T&E process in the DoD. An academic competency framework in the discipline of systems engineering would also have the impact on other government organizations and industry at large. It could accelerate the development of a pipeline for the next generation of a V&V/T&E workforce.

The framework will be grounded in DoD challenges that can complicate T&E. For example, the DoD test community often does not have insight or access into the embedded software on systems. This means that software testing is often black-box (i.e., testers can only set inputs and read outputs) or cannot be directly stimulated at all. Additionally, DoD systems are complex, leveraging a mix of technologies integrated onto a single platform, which raises issues of unit, system, and system of system V&V/T&E. The focus of T&E also changes throughout the system development lifecycle, which can be multiple years in duration. Early T&E often focuses on specific system technical and performance parameters and later testing focuses the operational capability that the system provides.

The research focuses on DoD needs, but we will also collaborate with the Department of Homeland Security (DHS), National Aeronautics and Space Administration (NASA), the International Test and Evaluation Association (ITEA) and others to ensure a broader perspective. Working with these additional organizations will improve the framework in two ways: 1) we may learn from them aspects that are relevant for DoD work, and 2) we can structure the framework so that it is easily adapted to fit other fields.

3.1.2 Motivation

Now is the right time to invest in a V&V and T&E competency framework. Complex military systems are rapidly evolving due to increased amounts of embedded software. Additionally, systems are already starting to incorporate machine learning (ML) and artificial intelligence (AI) algorithms to support varying levels of autonomy.¹ The system changes that result from increased embedded software require new V&V and T&E methods to keep pace. Figure 1 from the Defense Science Board (DSB) shows the evolution of embedded software in terms lines of code in fighter aircraft [1]. The figure shows a dramatic increase in recent years. The DSB highlights in this report the need for new T&E process for software, notably that a “software factory” is needed to rapidly test software changes.

The amount of embedded software in complex systems are pushing the limitations of current methods and processes for T&E. Statistical methods, including design of experiments, have provided foundational tools for T&E in recent years. The National Academies recommended these methods in 1998 [2], and they have been widely implemented in DoD testing. However, they are failing to provide the necessary tools for complex systems with embedded software in their current implementation. A key challenge is that embedded software enables capabilities in complex systems and test design approaches have revolved around modeling those capabilities across an operational space. For example, when testing a submarine with software updates, those software changes can provide indicators to improve operator's detection of potential contacts. This implies to test the system capability one needs to test the software against a range of potential contacts to ensure it provides the operator with useful, correct information. This is dramatically different from a hardware focused test that would have been conducted in the past that solely examined detection range as a function of the signature return. Software that can learn will further exacerbate this challenge.

¹ See for example General Dynamics Knifefish Unmanned Undersea Vehicle: <https://gdmissionsystems.com/en/underwater-vehicles/knifefish-unmanned-undersea-vehicle>

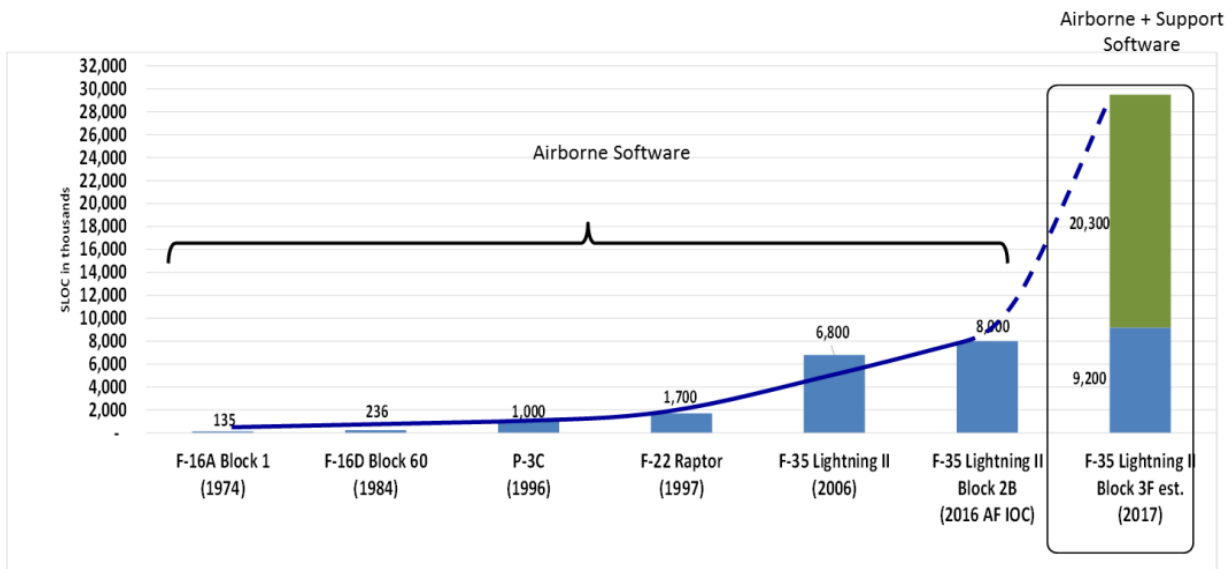


Figure 9. Software Lines of Code (SLOC) in Fighter Aircraft from the Defense Science Board's report on Design and Acquisition of Software for Defense Systems.

Another recent change that is driving the need for better V&V/T&E methods is the big data revolution. The product of any test is data and data-centric technologies are rapidly evolving. These evolving data-centric technologies include high performance computing power, distributed and sharable analysis methods, analysis specific software containers, cheap data collection and storage, and a better understanding of the physical world realized through models and/or simulations. These data centric technologies provide opportunities for paradigm shifts in V&V and T&E.

3.1.3 Framework Rationale

T&E / V&V are of essence to demonstrate systems are fit for purpose early on, in order to avoid problems once systems become operational. This is even more relevant with artificial intelligence-enabled cyber-physical systems. However, in academic disciplines we do not train people to do V&V / T&E, instead we focus on design. There are research papers about robust system architectures, about characteristics of good leadership, and what makes a good manager. However, it is much harder to define what a good T&E is. The limited resources that exist are often based in the needs of the DoD T&E community and not academically based. This is why we need this framework, to understand what makes robust and effective T&E, so that then we can guide the development of the T&E workforce to achieve better outcomes.

A framework will also enable increased speed in V&V and T&E method development. For example, the need for cybersecurity T&E has been a pressing issue in the DoD T&E community for many years now. The Director of Operational Test and Evaluation (DOT&E) first put out policy on the issue in 2012, a guidebook in 2014, and has continually updated guidance [3]. In spite of those efforts, the methods and process used to test cybersecurity in complex systems have been slow to evolve and most adversarial tests still rely heavily on adversarial red teams judgement of

the situation. Industry and the Federally Funded Research and Development Centers have reacted developing general cybersecurity test and evaluation frameworks, but they are disconnected from other test and evaluation frameworks (e.g., electronic warfare, radar, etc.) and lack academic review and input.

3.2 CURRENT STATE OF V&V AND T&E

3.2.1 Organizational-Centric Structures

Without a formal discipline around V&V and T&E, organizations need to develop their own approaches, practices, and policies. The DoD charges the Director, Operational Test and Evaluation (DOT&E) with developing guidance for all operational testing and the Developmental Test and Evaluation Office with developing guidance for developmental testing. Both organizations have generated substantial guidance around T&E, including what should be in Test and Evaluation Master Plans; how to conduct integrated testing; the use of statistical methods in T&E; and cybersecurity T&E guidance.² However, without an academic discipline to connect to the methods, the guidance developed has largely resulted in the development of case studies, which are not linked by an underlying methodology. Furthermore, the push for these methods can fluctuate with organizational leadership changes or be dominated by a single organizational perspective on methodology.

Another challenge is that organization specific guidance on V&V/T&E can lead to conflicting definitions between organizations on terms and approaches. For example, NASA tends to refer to the terms Verification and Validation when talking about system verification and validation, where the DoD uses them almost exclusively to discuss the verification and validation of computer models and simulations to ensure they are adequate for use in test and evaluation, training, etc. This confusion prevents rapid progress in method development.

3.2.2 Academic Discipline Frameworks

Many academic disciplines include elements of a V&V and T&E framework, key disciplines include: System Engineering, Human Factors, Statistics, Software Testing, and emerging research in machine learning and artificial intelligence assurance.

Statistical Frameworks

There has been an evolution of V&V/T&E methods for complex systems over the past several decades. In the Department of Defense, legacy test and evaluation methods focused on the use of military standards that dictated specific protocols, sample sizes, and test conditions. However, in recent years, the T&E community has adopted a statistical framework based in the field of Design of Experiments (DOE) for taking a process driven approach to T&E [4]. This methodology allows for test and evaluation practitioners to consider trade-offs in risk, sample size, coverage

² DOT&E guidance memos are publically available and located at: <https://www.dote.osd.mil/guidance.html>

of the test environment, and information required for decision making in a dynamic fashion. This methodology has proven valuable in testing complex systems that integrate technologies, humans, and missions, where outputs can be stochastic in nature.

The essential elements of an experimental design process are:

1. Identify the questions to be answered, also known as the goals or objectives of the test.
2. Identify the quantitative and qualitative metrics, also known as response variables, performance measures, or dependent variables.
3. Identify the factors that affect the response variables, which are also known as independent variables.
4. Identify applicable test design techniques.
5. Conduct testing
6. Analyze data
7. Draw conclusions

Sequential test design is a special class of experimental design where the planning phase remains the same (steps 1-3 above), but instead of selecting a design and then executing the test, the process is iterative where the next test point is based on a mathematical optimization across the desired planning criteria and data collected to date.

Scientific Test and Analysis Techniques (STAT)

STAT was defined by the DoD T&E communities to be a catch all phrase for all scientific approaches to collecting data when testing complex systems. Common methods and techniques that are included in this umbrella are design of experiments, observational studies (when control of test conditions is not possible), reliability data collection plans via operating characteristic curves, survey design and analysis for collecting human feedback, measurement system analysis (gage reproducibility and reliability), and wide ranging data analysis methodologies (various types of regression, Bayesian hierarchical models, etc.).

Uncertainty Quantification

Uncertainty quantification is an active field of research that attempts to estimate and ultimately reduce uncertainty based on multiple domains of information including computer models, simulations, live testing, and other sources of information. Computer models and simulations have the potential to provide critical information for testing complex systems. Models can be useful in bolstering conclusions for live testing by filling in gaps in knowledge, identifying edges of the performance space, and extrapolating findings to untestable conditions. However, formal methods for mathematically linking computer models and simulations to live testing and combining resulting uncertainties are lacking from current T&E practices.

Human Factors

Human factors testing is an important subset of experimental design worth highlighting because the methods may provide methodological underpinnings for testing any intelligent agents who, similar to humans, can learn over time. Human factors test techniques tend to fall into three broad categories: within-subjects designs, between-subjects designs, and split-plot designs,

which essentially vary how many treatment combinations an individual human is assigned in a test program and in what order they complete tests.

Software Verification and Validation

Software engineering delineates several levels of software testing: unit, integration, system, and acceptance. As changes are made to the software, regression testing can be performed at any level. This may include executing the original set of tests as well as adding new tests. Integration testing needs to consider interactions among components of the software. Testing all combinations of component levels in large systems is often infeasible, but the literature supports the use of pseudo-exhaustive testing as most faults in software are caused by interactions of few components (Kuhn 2004).

Formal methods testing frameworks are utilized when there is a syntax with mathematical semantics in which the software specification can be written and properties of the specification can be proven. One issue is that the specification may be verified, but the software implementation is not the specification and, as such, may have flaws. Axiomatic testing frameworks provide rules that work in the general case for software testing and should at least be considered. For example, two axioms of testing are that it is impossible to exhaustively test and, therefore, the most important tests should be run first. Testing should also be considered during the entire design process. Design for testability is the principle that features should be designed into systems so that the users can understand the system's intent by which decisions are made.

Emerging Machine Learning and AI Test and Evaluation

The current literature on test and evaluation of ML algorithms are heavily metric-based and do not revolve around standardized procedures. As ML algorithms and models become more complex, developing concrete metrics for evaluation is not straightforward because algorithm tuning is optimized directly based on test and evaluation metrics. Recent work V&V and T&E work has focused on understanding complex "black-box" ML models and evaluating how their complexity affects model performance. Ablation studies identify which mechanism/module within the larger algorithm actually improves performance. There is also test and evaluation research related to determining the vulnerabilities of artificial intelligence systems, specifically to synthetically generated inputs. Multi-agent systems, where each agent has knowledge of the other agents' states are becoming increasingly common. These autonomous agents dynamically adapt their autonomy during system operation. This research highlights the growing need to test and evaluate multi-agent systems in terms of not only their individual performance, but also their ability to seek the assistance of other agents.

3.2.3 Competency-Based Education

Given the complexity of the different contexts where engineers are required to implement V&V and T&E, we consider that using a competency-based educational approach will provide the required rigor for the development of these competencies while at the same time provide an innovative educational approach that will guarantee that students are able to transfer their V&V and T&E competencies into different contexts. Competency-based education (CBE) has been considerably growing in the last decades in higher education. Research on CBE suggest that this effective educational approach is desired by policymakers and industry leaders because it considers a holistic approach to educational development that is especially important because of the requirements of the contemporary workforce that include knowledge that goes beyond the development of technical skills. CBE promotes rigorous learning to prepare students for the workforce but also to become global leaders and citizens. This context-based approach simultaneously attends policy, practice, and community engagement. It goes beyond the traditional learning of technical or professional skills to a holistic approach that includes students' behaviors, contexts, skills, knowledge and previous experiences. In Figure 2 we present our CBE theoretical model including our educational products.

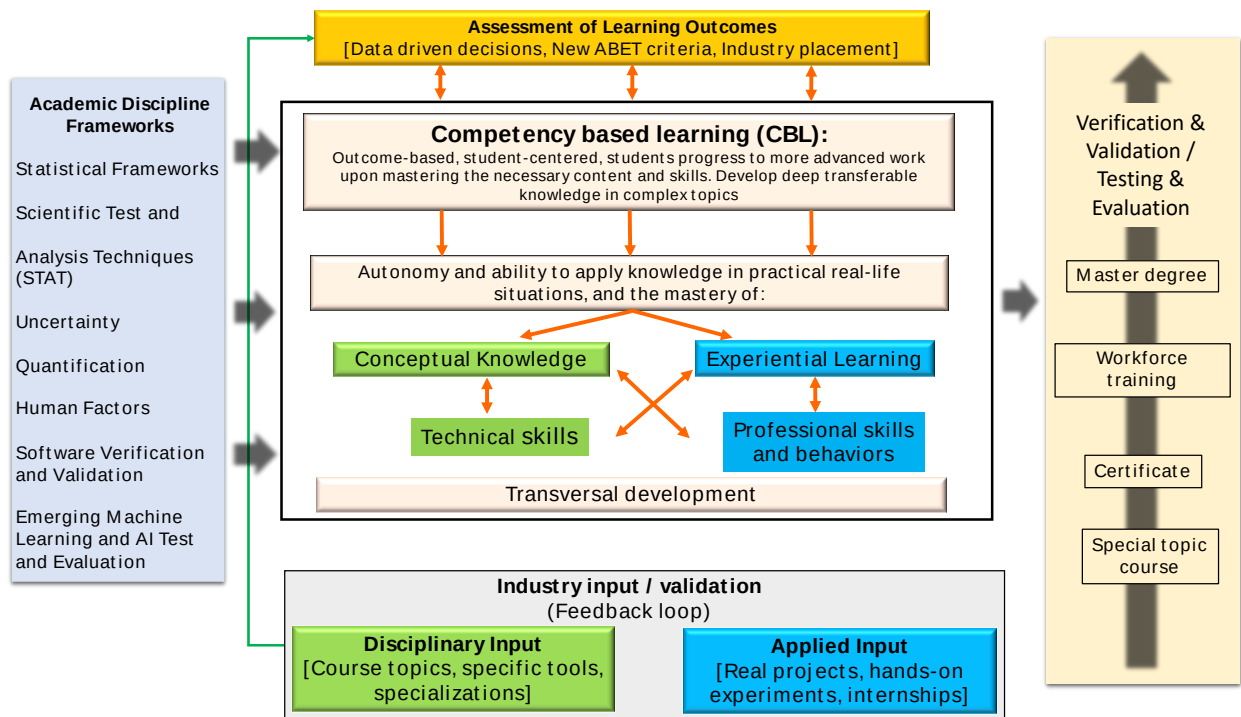


Figure 10. CBE Theoretical Model

3.3 RESEARCH PLAN

3.3.1 Proposed Work

The researchers will leverage their background in T&E, Systems Engineering, and Engineering Education to develop the competency framework. Leveraging the literature review on current competency frameworks for V&V and T&E performed under this incubator program, we will develop a competency framework that can be implemented to develop education and training programs, promote students that meet workforce requirements, and advance research.

3.3.2 Initial Framework Structure

The research team reviewed competency models from the DoD T&E community, developed initial core processes and competencies, and reviewed contributions from existing academic disciplines. All of these sources of information provide insights to a larger competency model.

T&E V&V competencies have been a crucial part of the following competencies models: International Council on Systems Engineering (INCOSE) [5], MITRE Systems Engineering Competency [6], Systems Thinking Enablers, National Aeronautics and Space Administration (NASA) Systems Engineering Competencies, Systems Engineering Competency Taxonomy, Generic Competency, Capacity for Engineering Systems Thinking (CEST) Competency, and Department of Defense (DoD) Engineering Competency.

Three existing competency frameworks were considered of most relevance: MITRE's Systems Engineering Competency Model, INCOSE's Systems Engineering Competency Framework, and the DoD's FY19 T&E Workforce Competency Model. Several V&V/T&E-related competencies were identified and classified (ref. Figure 11).



Figure 11. Survey of V&V/T&E-related competencies in existing competency models

In general, competencies were described as direct responses to required V&V and/or T&E tasks as part of a system development effort. For example, if the T&E task *Plan a T&E program* was necessary, then the framework or model would indicate that the T&E engineer must have the competency to *be able to plan a T&E program*. We find this approach to identify required competencies of little added value, since it essentially consists of repeating the V&V and/or T&E processes in a handbook in a different form. Following the previous example, the truly significant

question remains, *what competencies are necessary to plan a T&E program?* This is the question that drives our work in developing a competency framework for V&V/T&E.

In addition, most competency models/frameworks identified a transverse set of *professional* competencies (e.g., critical thinking, ethical reasoning, etc.) that would also be necessary for the V&V and/or T&E professional. Certainly, we agree that those competencies are likely necessary. However, once again, it is not evident from such categorization why those competencies are necessary or how they influence the success or effectiveness of a V&V/T&E engineer. This is another aspect that we consider critical in establishing a valuable competency framework for V&V/T&E.

In this incubator project, we have established a framework structure that focuses on the identification of competencies that will enable the execution of V&V/T&E tasks effectively and efficiently. The framework consists of two dimensions (ref. Figure 12). The first dimension addresses the type of work that the V&V/T&E engineer does. This is important because V&V/T&E tasks should drive the identification of competencies. Based on systems engineering handbooks and the competency frameworks listed above, we have defined four classes of tasks that V&V/T&E engineers perform: Strategic Planning, Tactical Planning, Execution, and Evaluation/Interpretation. They are described in

Table 1. The second dimension addresses the type of competency that is necessary to effectively and efficiently perform. We argue that, from a competency standpoint, adequately performing an engineering task requires competencies in exercising solid reasoning, effectively using available methods, and applying certain behavior. They are also described in

Table 1.

TASK CLASS	SPECIFIC TASK	COMPETENCIES		
		THINKING/ REASONING	METHODS	BEHAVIOR
STRATEGIC PLANNING				
TACTICAL PLANNING				
EXECUTION				
EVALUATION/ INTERPRETATION				

Figure 12. Proposed structure for a V&V/T&E Competency Framework

Table 1. Description of framework categories

Dimension 1	
Category	Description
Strategic Planning	In essence, all tasks that are necessary to decide what V&V or T&E activities have to be executed and when, as well as those that are necessary to anticipate courses of action as a function of the results of the execution of those activities, changes in the system development, or other unforeseen events that may affect the V&V and/or T&E plan. <i>Examples:</i> Define a V&V strategy, identify V&V risks, and establish verification criteria.
Tactical Planning	In essence, all tasks that are necessary to enable, plan, and prepare the execution of V&V/T&E activities. <i>Examples:</i> scheduling, facility selection and allocation, infrastructure identification and development.
Execution	All tasks that the V&V/T&E engineer performs during the execution of V&V/T&E activities. <i>Examples:</i> coordinate resources, manage risk, confirm validity of test configuration.
Evaluation/Interpretation	All tasks that are necessary to evaluate V&V results and interpret them in the context of the system purpose, system requirements, and stakeholder needs. <i>Examples:</i> determine data sufficiency, confirm fulfillment of requirements.
Dimension 2	
Category	Description
Thinking/Reasoning	Reasoning skills necessary to adequately use V&V/T&E methods. <i>Examples:</i> systems thinking, statistical reasoning, critical thinking.
Methods	Techniques, methodologies, and methods necessary to perform V&V/T&E tasks. <i>Examples:</i> regression analysis, inferential statistics.
Behavior	Behavior necessary to adequately, effectively, and efficiently leverage reasoning and methods competencies within a real V&V/T&E environment. <i>Examples:</i> self-discipline, attention to detail, facilitation.

By explicitly forcing to distinguish V&V/T&E tasks from competencies and distinguishing types of competencies, we suggest that actionable competencies can be identified. For example, consider

a test was performed a T&E engineer needs to evaluate the acceptability of the results (this is a T&E task). Keeping it simple, to be effective, a T&E engineer will need to be able to apply systems thinking to interpret results in the context of the system's purpose (Reasoning), be able to employ perhaps some statistical technique to make sense of the resulting data (Methods), be able to pay attention to detail to make sure that the test was conducted as planned in terms of configuration, steps, etc. (Behavior), and be able to not overstate the implications derived from the results (Behavior).

Similarly, by avoiding a category for horizontal professional categories and instead map all those competencies to specific tasks, we suggest that both identification of competencies and actionability of training and education programs can be improved. For example, consider ethical behavior as a required competency for a V&V/T&E engineer, as indicated by existing frameworks. A question remains, what is the meaning of being ethical? How do we train and/or educate an engineer to be ethical in the context of T&E? These questions can be answered by driving the identification of ethical behavior from the task in which the ethical behavior needs to be exhibited. Continuing with the example, consider the following two T&E tasks: Conduction of a test case and evaluation of test results. In the first task, ethical behavior is necessary to guarantee that all insights, non-conformances, etc. of the test are documented, as well as to make sure that the test strictly follows test requirements and established procedures. In other words, the ethical behavior in this case relates to follow established rules. In the second task, evaluation of test results, the implication of ethical behavior changes. In this case, ethical behavior is necessary to guarantee that the engineer makes an objective assessment of the meaning and potential consequences of the test results. While in the first case ethical behavior may be achieved by self-discipline (follow the rules), in the second one it may be achieved by self-confidence (ignore external pressures to inform the desired result instead of the actual result). Existing frameworks are unable to distinguish these subtle yet important competencies.

The research team's hypothesis is that a T&E V&V discipline should be framed within the domain of systems engineering and will draw from the fields of statistics, computer science, psychology, and other engineering disciplines. Additionally, the team has identified a preliminary structure for thinking about T&E foundational knowledge and T&E as a specialization. The figure below shows that structure. Just having V&V/T&E competencies along is insufficient to develop a test program. One must also understand how the operational use of the system and the system specific engineering domains to develop a credible T&E program. This research will identify critical connections to the domain knowledge and operational use case.

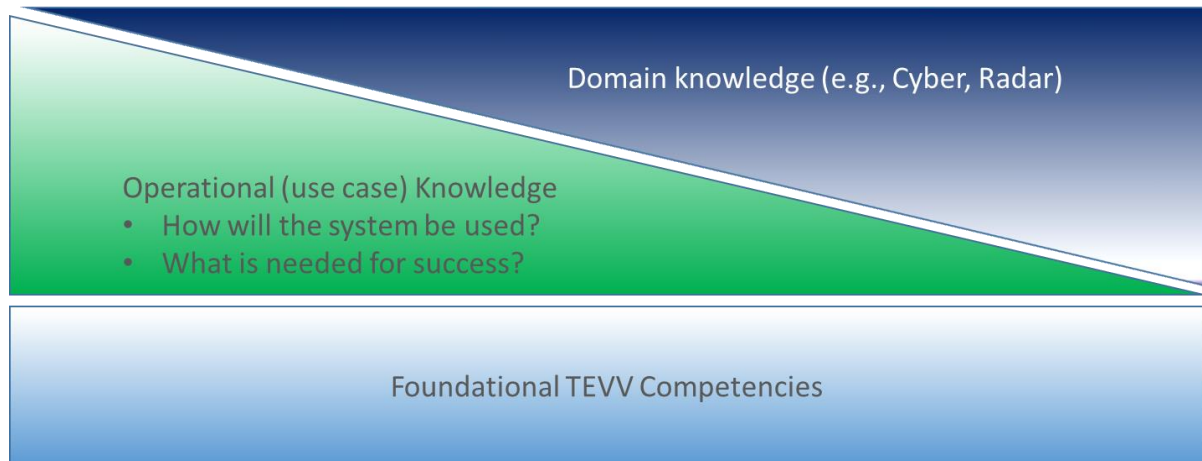


Figure 13. Relationship of Test, Evaluation, Verification, Validation (TEVV) Competencies to Additional Knowledge Needed to Scope a Test Program.

3.3.3 Methodology

The purpose of this project is to understand the adequacy and relevance of a competency-based V&V and T&E framework. Hence, we will leverage the discipline of engineering education and the development of new engineering curriculum to support our study. Virginia Tech has one of the top engineering education departments in the country, so one of our researchers is an engineering education expert. In order to promote a V&V and T&E competency framework that meets the complexities of the current engineering workforce it is important to include an industry perspective. The project will be conducted over multiple phases (see Figure 6), using data-collection methods such as interviews, focus groups, observations, and document analysis.

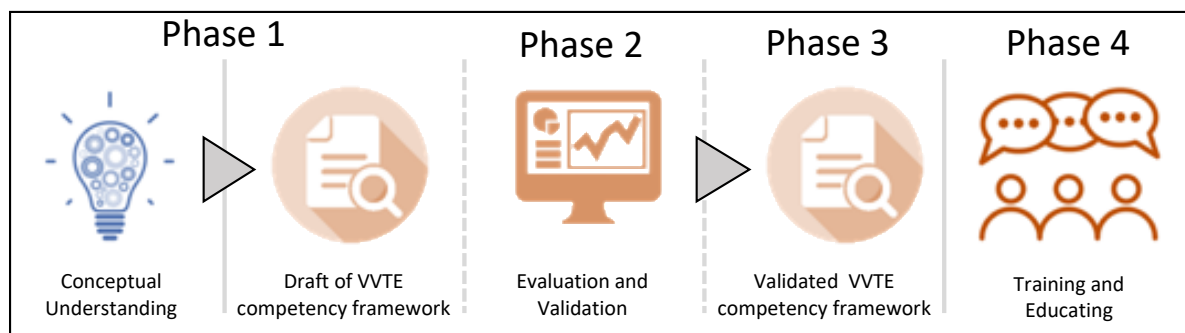


Figure 14. Phased Research Methodology

Phase 1 will focus on the development of our draft framework, which will be updated from this proposal. The framework will be developed by correlating and summarizing existing data sources: (a) conduct a V&V T&E process analysis, (b) identify existing systems engineering competency frameworks, (c) conduct a document analysis for syllabuses of existing courses in V&V and T&E, (d) identify information artifacts that are used or produced by T&E and V&V tasks (e.g., test plans, verification matrices), which may provide tangible insights into current capabilities and capability

gaps, and (e) conduct interviews with subject matter experts and members of the V&V/T&E DoD communities. The outcome of phase 1 will be a draft of the **V&V/T&E competency framework** representing a nuanced understanding of V&V/T&E contemporary requirements.

After developing an initial V&V/T&E competency framework, we then move toward measuring how good V&V/T&E engineers are fulfilling those competencies, or the second phase. **Phase 2** will focus on validating the adequacy of the framework. Following engineering education research best practices, we will develop a case study using a mixed methods approach to identify different data sources and assess the V&V/T&E competency framework. Data will be collected quantitatively by surveying experts both from industry and academia and ask them to rank the importance of the identified V&V/T&E competencies. In addition, we will survey students and have them identify V&V/T&E competencies. This will allow us to do triangulation of our findings and identify major incongruences or misconceptions that our initial framework can have. Then, qualitative data will be collected by conducting interviews and focus groups with the industry and academic experts, about the findings of the quantitative data and further explore their perceptions of the V&V/T&E competencies and any possible misalignment. These data will help us understand the alignment between the industry experts that are implementing V&V/T&E continuously and the academic experts that are in charge of promoting it. Finally, to validate the V&V/T&E competency framework we will use a sequential mixed methods approach to triangulate quantitative and qualitative findings and identify consistencies and misalignments. This information will help us update the framework (**Phase 3**) with the findings from phase 2 and put it in the context of existing systems engineering competency frameworks.

Lastly, **Phase 4** will focus on the development of educational interventions in order to reach a broader audience and have a positive impact not only in promoting V&V/T&E competencies, but in the overall workforce. Our CBE approach will allow us to develop curricular interventions that will guarantee that students and practitioners can be effectively trained to face the complex demands of the contemporary workforce. We aspire to develop courses on V&V/T&E, a graduate certificate, and a Master degree with the collaboration of industry partners that will be key to provide real contexts and projects that will simulate real-work situations. Furthermore, we aspire to develop workforce training that can be easily used in industry to contribute with the development of the workforce. Specifically, in phase 4 we would develop an overarching course curriculum that could be delivered as a special topics course at a University or used as the basis of a short course for workforce development. We would explore the potential for a Masters degree program with AFIT and NPS leveraging their existing curriculum and the results of this research. Finally, we would work with the DAU to implement any course development into the DAU T&E training courses.

Combined, these phases will lead to the implementation of a valid V&V/T&E competency framework that will be learned by students and will guide several curricular interventions.

3.3.4 Timeline

Gantt chart over two years key activities:

Phase	Task	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
1	Refine knowledge								
1	Refine Draft Framework								
2	Measure Framework Effectiveness								
2	Collect Data for Validation								
3	Perform Validation								
4	Establish Curriculum for V&V and T&E Professionals								

Program Effort: We propose a two-year program at \$200k per year totaling \$400k. The funds support the investigators, one/two graduate research assistants, and travel to meet with key organizational stakeholders in the test and evaluation community.

3.3.5 Audience and Impact

Numerous organizations are noting that their T&E workforce is not keeping up with the current change of system development. By developing a competency of T&E we can provide structure, advance research, and help re-train the current workforce. The following are specific examples of partnerships that could be established to maximize the impact:

- Air Force Test Pilot School (TPS) is expanding its curriculum beyond flight sciences into space, cyber. This research could better define the foundational knowledge and provide collaborations with TPS
- Air Force Institute of Technology (AFIT) and Naval Postgraduate School (NPS) both have curriculum in T&E, but it is not matched by other academic institutions. This research could strengthen their education offerings and provide organizational homes for academic curriculums outside of Virginia Tech.
- NASA, DHS, DOT&E, and DT&E are all noting that they have insufficient test procedures for advanced technologies (hypersonics, machine learning, artificial intelligence, 5G wireless). This competency model will provide a framework for educating the current workforce to develop new adaptive test methods for these technologies. Additionally, it could serve as the foundational resource for generating case studies in collaboration with these organizations to develop case studies that execute the competencies in the model, providing further validation to the model.
- The DoD has an Autonomy V&V and T&E community of interest, this framework could be leveraged by them to identify future research needs.

Additionally, the researchers have numerous related projects that would benefit from a competency framework these include:

- Performance Measures, Environments, Actuators, and Sensors (PEAS) Framework for T&E of Cognitive Agents. This research, led by Dr. Freeman and sponsored by the NRO looks at developing a T&E Framework for networks of cognitive agents in space. The

research is challenging because no definitive framework exists for T&E. Synergies with this research could provide a use case for examining the competency model.

- T&E for Project Maven. This research is led by Dr. Freeman and sponsored by Project Maven looks at how to develop test sets for machine learning algorithms before the algorithm is fielded. It also could provide a use case for the competency model connecting the specific algorithm testing into a larger framework.
- Enhancing Cybersecurity T&E. This research, led by Dr. Freeman and sponsored by DOT&E examines how cybersecurity T&E can be improved by developing and implementing best practices. It also examines how automation and machine learning can improve cybersecurity T&E.

3.3.6 Risks and Mitigation

Risk: Applicability. The competency model must be general enough for application to various defense systems. Previous T&E paradigms have focused on specific system T&E concerns, but failed to connect them into a larger discipline. However, the competency model must be specific enough to ... inform qualifications for hiring, develop educational materials (what are our specific aims)?

Mitigation: The research team will receive guidance, feedback, and peer reviews from a wide variety of stakeholders T&E related individuals across the DoD, DHS, NASA, and academic disciplines to ensure the framework is applicable.

Risk: Scalability. Across different system types and organizational structures.

Mitigation – use existing case studies and reference them against the competency model. Ensure reviewers have a range of system expertise.

3.4 EXECUTION STRATEGY AND DELIVERABLES

3.4.1 Research Team

Dr. Laura Freeman a Research Associate Professor and the Associate Director of the Intelligent Systems Lab at the Virginia Tech Hume Center. Her research leverages experimental methods for conducting research that looks at the intersections of cyber-physical systems, data science, artificial intelligence, and machine learning to address critical challenges in national security. She develops new methods for test and evaluation focusing on these emerging system technologies. Previously, Dr. Freeman was the Assistant Director of the Operational Evaluation Division at the Institute for Defense Analyses (IDA). During her tenure at IDA, Dr. Freeman participated in test planning activities for over 60 different DoD systems. She established and developed an interdisciplinary analytical team of statisticians, psychologists, and engineers to advance scientific approaches to DoD test and evaluation. During 2018, Dr. Freeman served as that acting Senior Technical Advisor for Director Operational Test and Evaluation (DOT&E). As the Senior Technical Advisor, Dr. Freeman provided leadership, advice, and counsel to all personnel on technical aspects of testing military systems. She reviewed test strategies, plans, and reports

from all systems on DOT&E oversight. Dr. Freeman has a B.S. in Aerospace Engineering, a M.S. in Statistics and a Ph.D. in Statistics.

Dr. Alejandro Salado is an Assistant Professor with the Grado Department of Industrial and Systems Engineering and the Co-Director of the Systems Engineering Program at Virginia Tech. His research focuses on improving problem formulation and V&V in large-scale cyber physical systems through the development of novel model-based approaches and quantitative methods. Previously, Dr. Salado was a systems engineer for over 10 years in the European space industry. During that time, he led and/or contributed to the development of space systems of up to \$1B, primarily in the areas of requirements, architecture, technical management, system integration, and V&V. In addition to several best paper award, he was the recipient of the NSF CAREER Award, the OAA Exemplary Dissertation Award, the Fabrycky-Blanchard Award for Systems Engineering Research, and the Fulbright International Science and Technology Award. Dr. Salado holds a B.S./M.S. in Electrical Engineering, a M.S. in Electronics Engineering, a M.S. in Project Management, a M.Eng. in Space Systems Engineering, and a Ph.D. in Systems Engineering.

Dr. Homero Murzi is an Assistant Professor in the Department of Engineering Education at Virginia Tech and leader of the Engineering Competencies, Learning, and Inclusive Practices for Success (ECLIPS) Lab. He holds degrees in Industrial Engineering (BS, MSc), Master of Business Administration (MBA) and in Engineering Education (PhD). Dr. Murzi 15 years of international experience working in industry and academia. His research focuses on contemporary pedagogical practices that promote industry-driven competency development in engineering, international engineering education, and finding effective educational interventions to prepare engineering students for the complex demands of the workforce.

Dr. Murzi industry experience focused on manufacturing engineering and talent management in different companies in South America. He was Talent Management Specialist for Johnson & Johnson, Consumer where he developed evaluation plans, and recruitment initiatives for employees in 5 different countries. In Academia, previous to Virginia Tech Dr. Murzi was an associate professor of Industrial and Systems Engineering at the National University of Tachira, in Venezuela.

Dr. Murzi has been recognized as a Diggs Teaching Scholar, a Graduate Academy for Teaching Excellence Fellow, a Diversity Scholar, a Fulbright Scholar and was inducted in the Bouchet Honor Society. He holds honorary positions at the University of Queensland in Australia and the University of Los Andes in Venezuela.

3.4.2 Deliverables

- Major Milestone reports (Phase 1-4)
- Competency model final report

3.5 REFERENCES

- [1] Department of Defense, Defense Science Board report on the Design and Acquisition of Software for Defense Systems. February 2018. Download available at: https://dsb.cto.mil/reports/2010s/DSB_SWA_Report_FINALdelivered2-21-2018.pdf
- [2] National Research Council. 1998. Statistics, Testing, and Defense Acquisition: New Approaches and Methodological Improvements. Washington, DC: The National Academies Press. <https://doi.org/10.17226/6037>.
- [3] Procedures for Operational Test and Evaluation of Cybersecurity in Acquisition, Director Operational Test and Evaluation. Available at: [https://www.dote.osd.mil/pub/policies/2018/20180403ProcsForOTEofCybersecurityInAcqPrograms\(17092\).pdf](https://www.dote.osd.mil/pub/policies/2018/20180403ProcsForOTEofCybersecurityInAcqPrograms(17092).pdf)
- [4] Freeman, L.J. and Warner C. "Informing the Warfighter – Why Statistical Methods Matter in Defense Testing." (2018) Chance Magazine 31 (2), 4-11. <https://chance.amstat.org/2018/04/informing-the-warfighter/>
- [5] INCOSE Competency Framework for Systems Engineering (2018). <https://www.incose.org/products-and-publications/competency-framework>
- [6] MITRE Systems Engineering Competency Model. Version 1.13E (2007). Download available at: <https://www.mitre.org/publications/technical-papers/systems-engineering-competency-model>

4. VALIDATION OF AI-ENABLED AND AUTONOMOUS LEARNING SYSTEMS – DR. PAUL COLLOPY, UNIVERSITY OF ALABAMA HUNTSVILLE AND DR. VALERIE SITTERLE, GEORGIA TECH RESEARCH INSTITUTE

4.1 OBJECTIVES OF THE RESEARCH

“The military’s current verification and validation process is meant for frozen software and is not suited to AIs that learn.” [Button, 2017] This research aims to create a method for validating autonomous systems, including those that learn from their experiences. New capabilities in autonomy, artificial intelligence (AI) and machine learning (ML) challenge the current Department of Defense (DoD) approach to system design and validation. It is difficult to trust a system without pre-defined behaviors, for which comprehensive testing of all possible states is impossible or infeasible. This evaluative intractability is a problem continuing to increase in magnitude and criticality with new advances in autonomy, artificial intelligence (AI) and machine learning (ML). Without a trusted method for validating AI-based, learning enabled systems, the DoD lacks the confidence and knowledge to use them effectively in theater.

The research described and proposed here, aims to **develop a method for validating the spectrum of AI and learning enabled systems through the use of a formal, logical argument methodology**. The concept of argument analysis is new to most of Systems Engineering (SE), especially those areas focused on technical design and validation of that design. Yet when traditional deterministic analyses and decomposition approaches fail, namely when faced with uncertainty and a problem state space nearly mathematically impossible to bound, argument analysis is well-suited to support decision analysis. Jennifer Stevens at NASA / MSFC is using an argument-based approach to develop a new formalization of SE validation particularly applicable to autonomous and adaptive systems. Her concept replaces confirmation of static behaviors with a structured logical argument of capability success in the field. [Stevens, 2017].

Using Stevens’ work as a conceptual foundation, this effort seeks to develop a formal argument process that may be codified in a user-guided toolset specifically designed to support validation of AI and learning enabled systems. The research aims to develop the methodology to provide a logical basis of behavioral validation when supported with the data the argument identifies as necessary to support its claims across a spectrum of differing AI and ML capabilities. The penultimate goal of this effort is to create a prototype tool using the method that enables validation of AI systems in a logical, consistent, non-*ad hoc* manner throughout DoD Acquisitions. Succinctly, the aim is to enable the DoD to field such systems faster and with greater trust in their behavioral characteristics through a formal, repeatable, understandable approach guiding these systems through the acquisitions process to the Warfighter.

4.2 CURRENT PRACTICE AND ITS LIMITATIONS

Kinetic activity in theater is all about decision-making, and AI is a powerful decision-making technology. Program Executive Officers (PEOs) and Warfighters responsible for introducing technologies to the field capable of closing the OODA loop faster than our adversaries' while also requiring trust in those capabilities, need the adaptability, flexibility and responsiveness that only AI and autonomy can deliver. The need for a new method to validate these systems was emphasized by several DoD engineering centers in the SERC Pathfinder study [Verma, 2018].

The traditional approach to validation testing, one that hopes to exhaustively evaluated the entire possible problem space, has become hopelessly impractical even for conventional (non-AI) complex systems [Felder and Collopy, 2012]. This approach is even more obviously inadequate for a system that formulates its own actions after development is complete. This is exacerbated by the inherent non-stationarity of learning enabled (LE) systems. Behavior of LE systems will be as dynamic as the field data they ingest, so that a more dynamic, life-cycle pervasive concept of validation is necessary. As highlighted in the 2010 Air Force Technology Horizon report [Dahm, 2010], achieving gains *"...from use of autonomous systems will require developing new methods to establish "certifiable trust in autonomy" through verification and validation (V&V) of the near-infinite state systems that result from high levels of adaptability; the lack of suitable V&V methods today prevents all but relatively low levels of autonomy from being certified for use."*

This same conundrum is found in software validation. Traditional approaches used a decomposable tree to capture the relationship between decision states and algorithmic sequences. For even basic ML, however, this is not possible; data is ingested by the algorithm and then, in effect, decisions that are highly inter-related take place within a virtual black box to produce the final result. DARPA's program on *Explainable Artificial Intelligence (XAI)*, striving to develop approaches whereby AI/ML systems *"have the ability to explain their rationale, characterize their strengths and weaknesses, and convey an understanding of how they will behave in the future"* arose from this very problem [DARPA, 2016]. Ironically, the program sought primarily to develop novel technologies to make the AI technologies explainable, in particular, to develop new, explainable machine learning techniques. More recent forays such as DARPA's *Assured Autonomy* program [DARPA, 2017] seek to bound the space for LE systems by presuming a definable ground truth exists (i.e., AI/ML for navigation or robotics control for which there are discernible bounds of desirable versus undesirable state spaces). Yet, the real need is to enable systems to operate confidently in less rigidly defined environments, without predetermined constraints. The most powerful AI/ML will learn in theater to identify and/or classify threats, determine firing solutions, respond to evolving enemy tactics (possibly driven by enemy ML systems), or otherwise actively engage in control operations and decision-making. In these cases and others, data in the field is dynamic, uncertain, and often looks nothing like training data.

Many research efforts have addressed particular aspects of the problem space through stability theory or variations on the notion of intelligence testing stemming from Turing. As an exemplar of the former, [Zhang et al., 2015] consider verification of a class of closed-loop systems

containing a neural network controller for first-order, nonlinear systems with uncertainty. In their work, the closed-loop system may face faults in either the cyber or the physical system components, i.e., the control software or the physical portions of the system engineered to produce the manifested capability respectively. Zhang et al. developed an approach based on Lyapunov stability theory to detect unstable learning behaviors due to unanticipated faults within the cyber and/or physical portions of the system. The work focuses on a critical part of the overall problem space, using unstable learning behaviors as a proxy alert for undesirable dynamics and hence compromise to safety. It is, however, inwardly focused as part of system verification (not validation); the approach considers the system only and so does not move toward a methodology that would provide confidence in system behavior under potentially highly-varying field conditions.

[Li et al., 2018] evaluate perceived relationship between AI and intelligence demonstrated through designed tests. The fundamental assumption captured in the state-of-the-art review is that solid performance via intelligence tests will demonstrate appropriate, reliable behavior. (This presumption arguably does not even hold with humans, despite the intent to design machine intelligence to approach human performance.) Li et al. observed the traditional SE V-diagram results in design of system testing plans prior to actual coding development of the system itself, and so proposed a different, parallel loop process of test specification and design with simulation. While requiring significant human involvement to accomplish, Li et al. noted this is not substantially different from many state-of-the-art approaches, including the Intelligent Vehicle Future Challenge designed by Tsingua University and Qingdao VIPioneers that used human judges to evaluate autonomous vehicle performance.

[Roff and Danks, 2018] discuss the notion of trust for autonomous systems, recognizing that trust is a complex, multi-dimensional notion and not a binary concept. There are varying levels of autonomy, each affecting the development of trust differently, and each posing enormous, albeit varying, challenges to testing, verification, and validation. Trust, while related to predictability and reliability, which are characterized through patterns of behavior, also contains a psychological aspect, particularly when trust across manned-unmanned teaming is considered. The authors consequently advise against relying on binary “yes, trust” or “no, do not” frameworks and methods to discern actionable guidance. They similarly have little faith in purely technological solutions. Building from the discussion of Roff and Danks, [2018], a methodology – such as the Toulmin argument approach proposed here – that builds confidence in answering the question “What would it take to convince us that X is true?” addresses the psychological aspect of human belief within the critical attribute of trust.

Software engineering has long struggled with the combinatorial explosion of the possible state space of scenarios and/or parametric combinations for verification and validation. Only very simple task spaces may be enumerated via brute force, so sampling of the possible scenario space has become a common approach. Sampling leaves considerable gaps, however, and newer methods such as satisfiability, already used for hardware verification, are advancing into practice. Huele and Kullman [2017] review the rise of satisfiability solvers (SATs), which still require the use of supercomputers, that add “brute reason” to “brute force” exploration of a search space.

SATs determine whether a set of sentences in Propositional Logic (sometimes called first-order predicate logic) is satisfiable. The solution is achieved by reducing the complete search space through various reasoning heuristics or backtracking search methods [Huele and Kullman, 2017; Stanford, 2019]. More recent SAT techniques evolved out of efforts to prove unsatisfiability as opposed to proving verified satisfiability of complicated software. Even so, SAT proof of the Boolean Pythagorean Triples problem required 200 terabytes, a huge growth over the previously largest mathematical proof, which was 14-gigabytes in size [Lamb, 2016]. Even 14 gigabytes constitutes a proof that is too large to be read by a human. SAT at this scale, therefore, must rely on other “trusted” algorithmic systems for validation of the problem or system in question.

[Boehm, 2006] discusses the parallel evolution of software engineering and systems engineering away from sequential, reductionist methods toward “softer” processes emphasizing a more spiral, continuous learning approach. Boehm observes that as it becomes clear that software requirements emergence is incompatible with traditional, sequential, waterfall process models, *“Fundamentally, the theory underlying software and systems engineering process models needs to evolve from purely reductionist “modern” world views (universal, general, timeless, written) to a synthesis of these and situational “postmodern” world views (particular, local, timely, oral) as discussed in [Toulmin, 1992].”* This is congruent with the philosophy espoused in the current work and its proposal to extend Toulmin’s argument as a “situational view” of validation for LE systems distinct from what traditional, decomposition, or deterministic techniques can offer.

4.2.1 Validation

The SEBoK defines System Validation as *“a set of actions used to check the compliance of any element ... with its purpose and functions”* and focuses on checking conformance to system requirements. It stresses that validation is a process that should occur throughout the lifecycle, but too often validation is lumped with verification and performed as the last step prior to system delivery. When the intended functions of a system can be completely spelled out in advance, this approach to validation has often been successful. However, the whole reason for autonomous, adaptable, augmented-intelligence systems is to act in ways which were not specified by the designers. Thus compliance with functions is too narrow a validation to create trust in a learning system.

Validation is related to but distinct from assurance, which started out meaning that the system is not broken (i.e., quality assurance), before migrating toward a NASA-type view that the system is safe. This expanded to mission assurance, a guarantee that the system will not fail to execute the mission, which becomes a lot like the original meaning of validation, except for the subtle distinction between system success and avoidance of system failure. One can imagine a system that defeats a peer enemy in four out of five engagements (it is valid), but breaks in two out of ten engagements (it is not assured). Assurance has branched out to cover cybersecurity, protection of internal information from disclosure, and other dimensions of system safety.

Definitions in standards for validation have been regressing for the past 15 or 20 years toward an industry-preferred notion of proof that the system meets its requirements. This creates a lot of

gray area and terminology conflation when seeking to separate validation from verification, which already means proof that requirements are met. (As per SEBOK, verification is “*the confirmation, through the provision of objective evidence, that specified requirements have been fulfilled.*”) Validation is supposed to mean, ultimately, that the system is successful when deployed. That is, however, a high bar – one that is increasingly difficult to prove for AI and LE systems. Success entails many factors outside of industry's control. **The research team for this effort posits that validation, prior to fielding, is knowledge supporting the belief and argument that the system will be successful in the field.**

4.2.2 AI/ML Classification, the Defense View, and Adaptive Systems in the Field

There are two primary types of ‘machine intelligences’ used today for decision support: expert systems and machine learning. Expert systems, as rules-based algorithms developed by subject matter experts (SMEs), were one of the first forays into artificial intelligence development and are the dominant structure used in many commercial and Defense applications. These systems struggled to keep pace with ever-expanding possible scenarios, however, and so machine learning began to take hold as part of the second development wave. While there are a number of machine learning techniques, neural networks (NNs) have become ubiquitous in web-based commercial software. Unlike expert systems, NNs can be quickly retrained to adapt to new circumstances. This is a critical point, however, as a ML algorithm’s internal model is limited to the data used to train it. Despite being flexible and computationally powerful, NNs are also black boxes; their internal models are inscrutable. Consequently, there is no way to predict ahead of time how a given ML algorithm will react to a novel input or how the algorithm will behave after ingesting copious amounts of new data over time. This lack of transparency is a critical impediment to trust in ML-based systems, especially those that continue to learn, for use in high-stake military engagements and was the genesis for the DARPA XAI program discussed in Section **Error! Reference source not found..**

There are a multitude of ways to classify AI, ML, and LE systems including how they learn (with supervised and unsupervised characterizing differing degrees of autonomy), the breadth of what they are designed/expected to learn, how they function (e.g., via regression, clustering, dimension reduction, neural network based structures, etc.), or their intended functionality (e.g., image classification, decision recommender systems, natural language processing, etc.). Not everything across this spectrum of capabilities and varying degrees of autonomy is classified as learning. Narrow AI/ML logic is frozen after being trained on data while other AI is not frozen but limited in behavioral complexity, for example. The current research is grounded in the understanding that there is (a) a spectrum of AI/ML capability for Defense systems, illustrated in Figure 1-1, and (b) LE systems are required for robust operations adaptable to the dynamics inherent in theater.

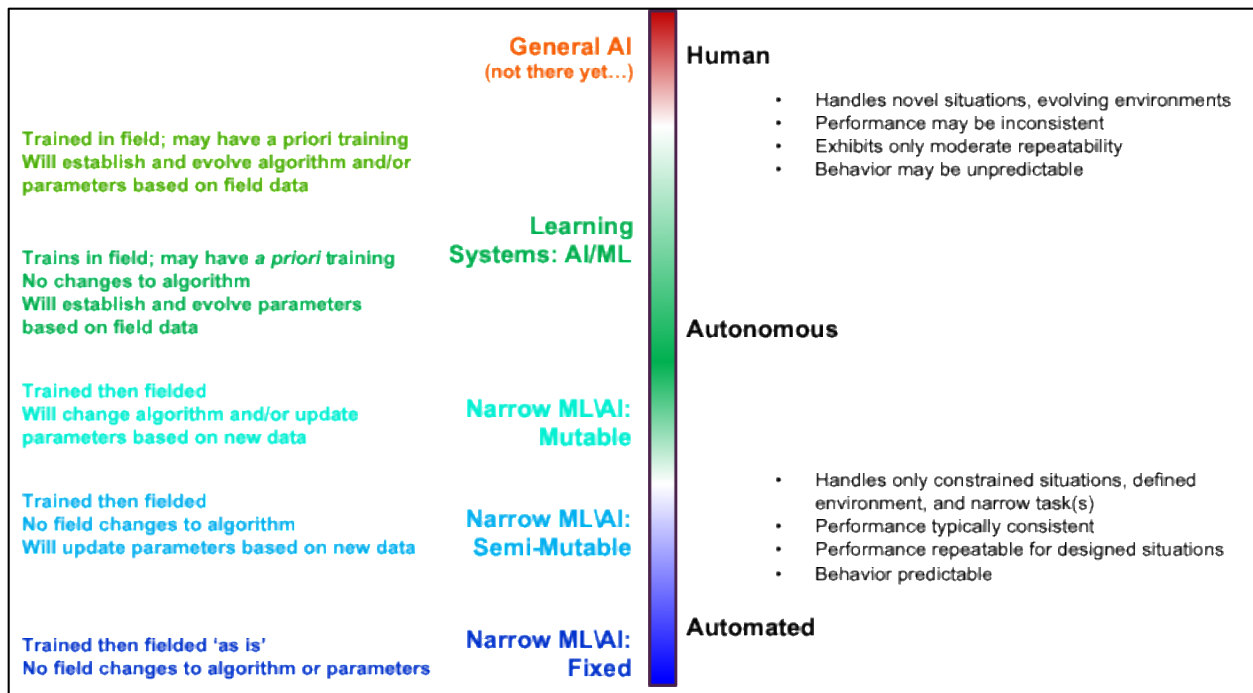


Figure1-1: The spectrum of Machine Learning (ML), Artificial Intelligence (AI), and Learning-Enabled systems for validation considerations

Narrow AI\ML is typically task-specific and fixed in terms of its algorithm code and parameters once trained. These systems exhibit the same brittleness and systems considerations as other automated engineering systems. The momentum is solidly toward systems that are able to handle a greater number of operational situations with a greater degree of autonomy. The increase in potential capability manifests as increased algorithmic complexity and hence reduced understandability and predictability of system performance as discussed above. It also brings potentially increased vulnerability though increased attack surfaces, especially if manned-unmanned teaming is required to achieve operational performance. As the development goals push toward increasingly autonomous, learning systems – ones more “human-like” in adaptive capabilities – the traditional culture within the intersection of the DoD and SE communities still desires predictable, guaranteed performance. This standard is beyond what can be promised even for highly trained humans. A paradigm shift in how we approach validation and develop trust in system readiness for fielding is necessary.

There are practical considerations for DoD use of learning AI\ML systems as well. For example, when may they be fielded? When is training sufficient for real operation, and when must this occur prior to fielding verses when learning in the field is necessary to exploit contextual and changing data? Many current programs still use expert-curated, cleansed, and validated data sets to train ML systems, ensuring they are trained within stable, anticipated bounds. Data integrity cannot be guaranteed in the field. If a system is adapting – learning – in the field, how much of this process could or should be automated (i.e., trusting the system to learn autonomously in addition to operating with some degree of autonomy)?

Especially vital for the DoD to consider is the notion of asset transferability. With ML capabilities and performance driven by data, it is critically important that data in Area X is often very different than data in Area Y. An AI\ML algorithm trained on facial or vehicle recognition in Atlanta, for example, will face a very different data environment if suddenly transplanted to Mosul. LE system performance in one theater of operations cannot be expected to be preserved in another theater of operations, at least until some initial re-learning period of time has passed. This challenges the very concept of a Digital Twin. Without the same data, a digital twin model could not reproduce the operations of a LE system, much less the operations of multiple instances of the same initial LE system that continued to learn in their respective Areas of Responsibility (AoRs).

Far from the pipedream of systems engineering certainty, the DoD has its own historical ways of dealing with extreme uncertainty in operational scenarios and combinatorically intractable problem spaces. The Defense Advanced Projects Agency (DARPA) Grand Challenge 2005 (DGC05) and Urban Challenge 2007 (DUC07) for autonomous ground vehicles (AGVs) took an integrative approach. Instead of utilizing point-testing of the various individual components or algorithms that are required to compose a fully autonomous vehicle, DARPA designed one massive test, requiring a full-system solution. DGC05 and DUC07 each required an AGV to perform a multitude of tasks (e.g., perform complex maneuvers, traverse rugged terrain, drive in traffic, recognize and manage underpasses, etc.) to complete the one test. [DARPA Prize Authority, 2006; DARPA Urban Challenge, 2007]. While not comprehensive in terms of an entire range of possible scenarios, the difficulty of the competitive, integrative model advanced the field considerably; several other DARPA programs from robotics to cyber went on to use the comprehensive challenge model [DARPA, 2014].

Similarly, the DoD relies on multiple, large-scale simulations of tactics and operations such as One Semi-Automated Forces (OneSAF) and Combined Arms Analysis Tool for the 21st Century (Combat XXI). These frameworks implement higher fidelity representations of platforms, soldiers, equipment, logistical supplies, communications, and other actors and effects within the operational environment. More detail is equated to more realism. It takes significant time to set up and modify a particular scenario for simulation which, even though the output is stochastic, is still just a single vignette. Often, entity behaviors in these simulations are programmed at such a high level of fidelity that they are not reusable in other scenarios. In a recent report by RAND [Wilson, 2018], model validation is viewed as the degree to which the M&S reflects the “real world”, with conceptual validation evaluating if the individual model components adequately represent their real world counter parts (e.g., weapons systems, units, etc.) and output validation relying on comparison of M&S results to data obtained from another “credible domain”. The credible domain, defined in one US Army publication cited by RAND, is presumed to be the real world or a “source that is recognized as an expert on the relevant characteristics of the real world.”

Despite striving for fidelity, human behavior models in these scenarios, as well as the complete variation of a state space for physical parameters and their complex relationships, are simplified by necessity. Decision making by entities in such models is highly limited and pre-defined, which

is not an accurate guide to decision making in an actual battlefield. The DoD consequently holds that these force-on-force M&S frameworks produce insight, not concrete truths in terms of outcome specifics. There may be some limited circumstances in which a combination of scientific theory, accepted algorithms, and SMEs may be able to generally validate the structure of a M&S scenario and its approximate realism. It is not, however, absolutely certain. The DoD therefore already accepts a degree of risk in its acceptance of highly complex system analysis with a necessary lack of complete coverage of the possible state space.

4.2.3 The Toulmin Argument and its Applicability to AI and LE System Validation

Stephen Toulmin was a British philosopher who began his working life during World War II in a role that now would be called a radar systems engineer. After the war, at Cambridge, he studied under Ludwig Wittgenstein who strongly influenced Toulmin's search for truth in a complex and contextual world. Toulmin found that the standard for truth in philosophy was often Propositional Logic, which has been very successful as a truth standard in mathematics. However, this standard was seldom useful outside mathematics because Propositional Logic is too brittle and too restrictive.

Propositional Logic is based on the syllogism, summarized formally as *(A and A implies B) proves B*. An example might be "Bethesda is a city, and if something is a city, then it must contain buildings, therefore Bethesda contains buildings." The positive aspect of the syllogism is that, when it can be used, it is very reliable. The negative side is that, in contextual reasoning about real-world situations, syllogisms can prove very little beyond what is obviously true in the first place.

To address this shortfall, Toulmin developed a softer and more robust deductive logic that has since found application in many disciplines [Toulmin, 2003]. Figure 1-2 (adapted from Erduran, Simon and Osborne, [2004]) shows the basic elements of Toulmin's logic. He begins with a claim, a statement that might be true or false, and we would like to know whether it is true. The claim is supported by data drawn from the world or from experience. So far, the claim replaces B in the syllogism, and the data replaces A. Toulmin creates a warrant to replace "A implies B." The warrant shows how the data lead to a conclusion that the claim is true. The strength of Toulmin's system is that the warrant can employ practical reasoning and common sense. It is not restricted to sterile statements such as "A implies B."

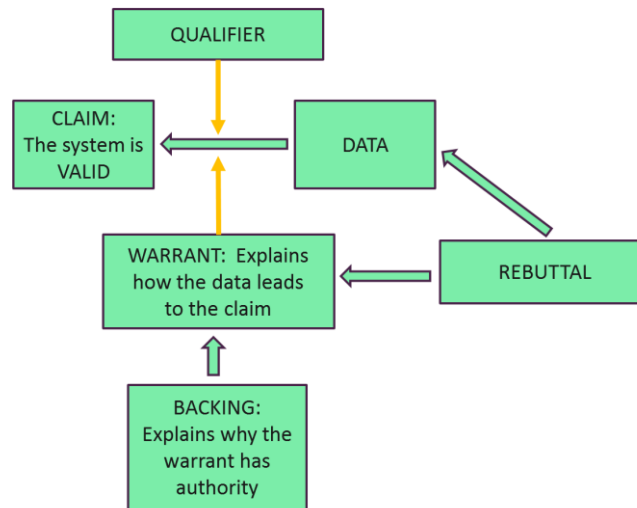


Figure 1-2: The Toulmin model of soft logic

Other elements support or clarify the relations between data, warrant, and claim. The backing is a way to reference a library, authority, or experience that lends credibility to the warrant. Rebuttals are used to restrict the applicability of data or warrants – the rebuttal identifies areas where the data does not apply or the warrant is not legitimate.

A major difference between Propositional Logic and the Toulmin model is the qualifier. In Propositional Logic, any statement can only possess one of three qualities with respect to truth: the statement is true, or false, or undecided. The Toulmin model, in deference to actual language and the nature of fact in the real world, allows all levels of qualifiers. Statements can be possible, or likely, or almost surely true, or most definitely true. Qualifiers can be applied to data, warrants, or claims to indicate the extent to which they are true. When Bayesian reasoning is available, qualifiers can be numerical probabilities, and in some cases this will allow the qualifiers of various data and warrants to be mathematically combined to form the qualifier for the claim, which is a form of uncertainty quantification.

AI began with algorithms that implemented Propositional Logic, so it is not surprising that Toulmin's model has also spawned a class of AI algorithms. Work in AI has pursued two main directions:

- (1) Abstract argumentation considers each argument as an atomic entity without internal structure. It provides a framework to model and analyze attack relations between arguments and sets of arguments, and develops supporting instrumentation. Dung [1995] defined the base representation (the argument framework diagram) and key properties (conflict free arguments, admissible arguments, provable and refutable arguments). Jakobovits and Vermeir [1999] and Caminada [2006] define an algorithm that assigns label to arguments within such diagrams (defeated, justified according to the topology of attacks relations).

- (2) Structured argumentation examines the internal structure of arguments to draw conclusions. It provides methods for extracting such arguments (often from text) and for comparing their support and attack components. Comparison principles concern conclusive force, such as specificity (the more specific argument wins), basic order (strict arguments win over defeasible arguments); number (arguments with fewer defeasible steps are stronger), and weakest link (arguments are no stronger than their weakest component, and the argument with the strongest weakest link wins). Note that the core assessment of conclusive force comes from contextual knowledge external to argumentation theory.

Automated methods combine support and attack relations to draw conclusions. For example, the ASPIC+ system [Prakken, 2010] maintains support and attack arguments as separate entities, then assigns labels to arguments (warranted, undermined, undercut, rebutted, defeated) by reasoning over the assemblage (Figure 1-3).

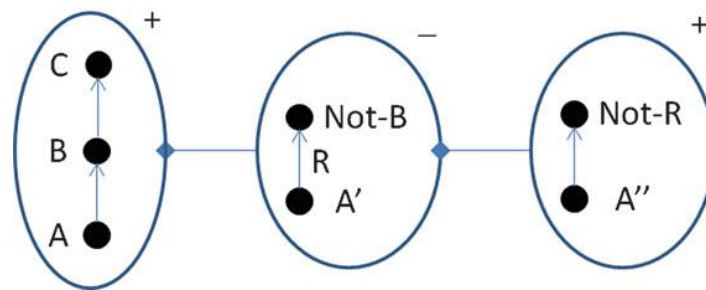


Figure 1-3. ASPIC+ Logic Diagram

The bulk of the AI work in argumentation theory has been associated with legal reasoning and dialogue analysis using structured argumentation models (see, for example, the COMMA conference - Computational Modelling of Argument, and the journal *AI and the Law*). There are quite a few implemented systems, which may be useful to systems engineering research either as metaphors, or cranks for extracting and analyzing validation arguments. The Handbook of Argumentation Theory [van Eemeren et al., 2014] identifies argument diagramming software, argument evaluation software, the Argument Markup Language (AML) and Argument Interchange Formats [Chesnevar et al., 2006], in addition to the more familiar mechanisms of Bayesian epistemology that assess argument strength as $p(\text{hypothesis} \mid \text{evidence})$.

4.3 RESEARCH APPROACH

The challenge in this research effort is to **define a methodology and process that is formal, hierarchical, and repeatable**, based in the fundamental concept of validation as collection of knowledge to build confidence that a system enables a capability or creates a desired effect in the world, such as winning a military engagement. Confidence comes from an argument that the system will enable the capability and create the effect. This argument starts at the very beginning of concept development (why do we believe this design is a good idea?) and grows throughout

system definition and realization, becoming more compelling in each step (which meets the SEBoK goal of a validation process that extends across the entire development process). As described in Section **Error! Reference source not found.**, the structure of the argument is created as a part of initial feasibility studies to enable reasoning about system validation and provide grounds to justify or refute heuristics. Throughout development, the validation process collects, adds and limits rebuttals, and builds the warrant.

In terms of impact, the team seeks to develop a holistic framework to guide validation of systems across this capability spectrum, as opposed to one-size-fits-all. The team believes an emphasis on “What should validation assess and measure?” instead of “What evaluation tool can I build?” will provide the critical, necessary focus to ground this work and future efforts that benefit from such definition. Similarly, the research recognizes that different AI/ ML implementations will exhibit different behavioral characteristics and thereby require differing argument structures; this understanding will also guide the development.

4.3.1 Phase 0 – Pilot Background Review and Foundational Concept Development

For the incubator phase, the research team took a multi-faceted approach. First, the team began to evaluate existing paradigms for LE or complex systems that are similarly analytically intractable to understand existing notions of what can be validated and acceptable levels of risk. Namely, these included autonomous vehicles and operational simulations such as OneSAF and COMBAT XXI. This allowed for clear distinctions between trained-and-fixed ML and adaptive or continuously learning AI/ML and a consequent focus for the current research. Specifically, the effort sought to understand the spectrum of AI capabilities relevant to validation as a framework, evaluating what may serve as an argument basis in each, and begin to fill in the components of the Toulmin Argument Pattern. The primary focus of the argument-based methodology is the shaded region of that spectrum as shown in Figure .

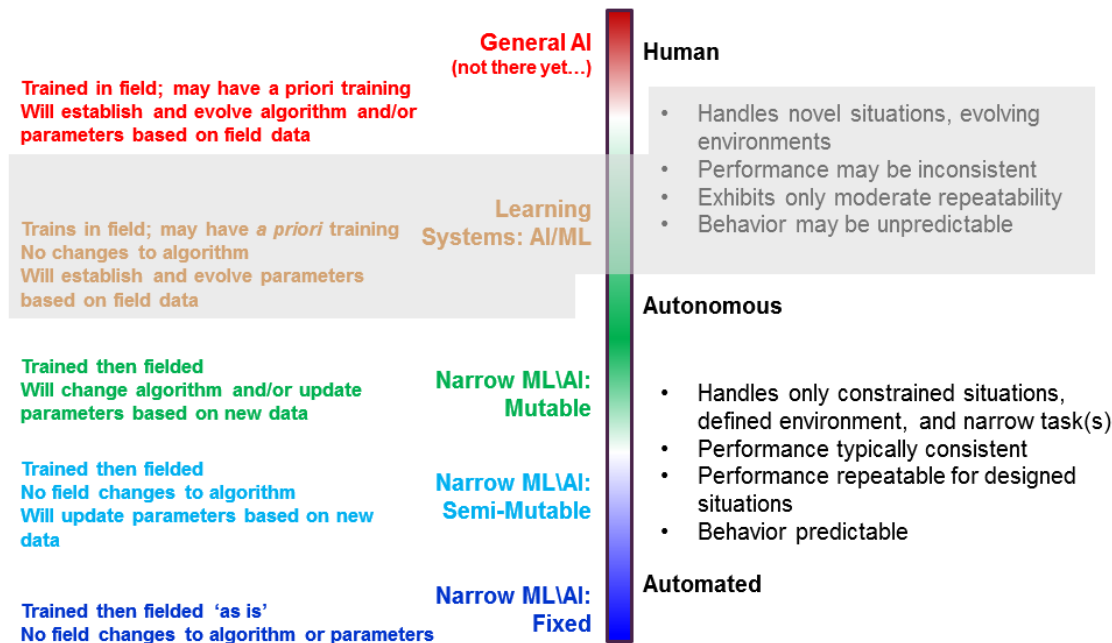


Figure 1-4: AI\ML to LE system capability spectrum with shaded region representing research focus

Figure 1-5 shows a preliminary application of this method to a Venus autonomous lander, illustrating our prototype tool, which is based on Vue.js, an open-source Java-based framework. The claim is that the Entry-Descent-Landing phase of the lander will be successful. Note that the claim is supported without reference to functional requirements. One of our goals was to show that validation can be performed in the absence of functional requirements, a necessary step to validate field-adaptable systems, where the future set of functions is unknown at design time.

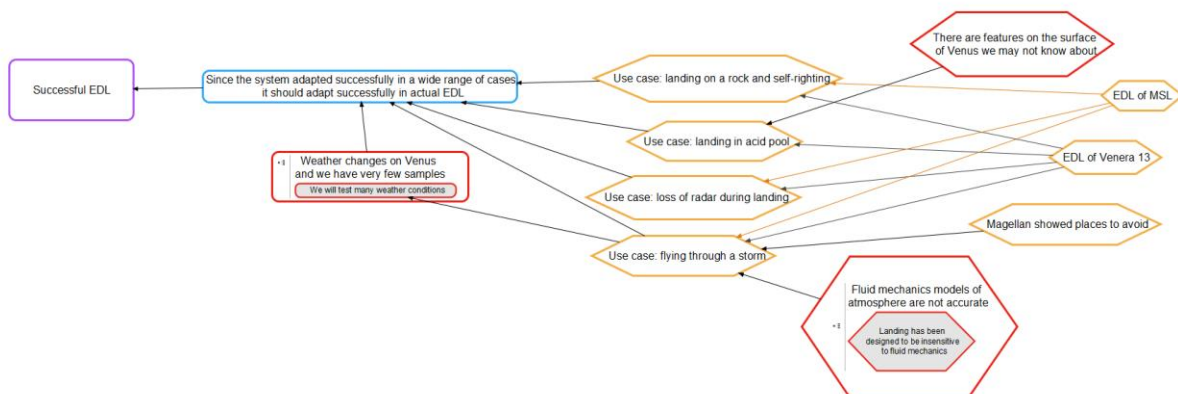


Figure 1-5: Validation of Entry-Descent-Landing for an Autonomous Venus Lander. This validation is just a partial example, it is not intended to provide complete coverage.

The encoding of the Toulmin model elements, detailed in Figure 1-6, is as follows: Claims are in purple boxes, warrants are in blue boxes, and data are in yellow hexagons. Qualifiers are outlined in black, taking the shape of the element that they qualify. Rebuttals are outlined in red, taking the shape of the element they rebut. Rebuttals may contain a rebuttal to themselves, a statement about where the rebuttal does not apply.

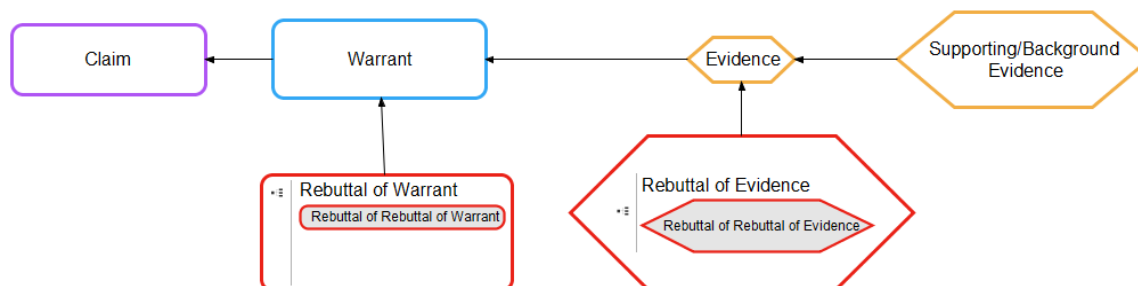


Figure 1-6. Notation in our tool of the Toulmin model elements.

Figure 1-7 shows our second prototype application, an Unmanned Ground Vehicle (UGV) that might perform reconnaissance. Again, this is incomplete, but developed as an example to assess basic feasibility of the method.

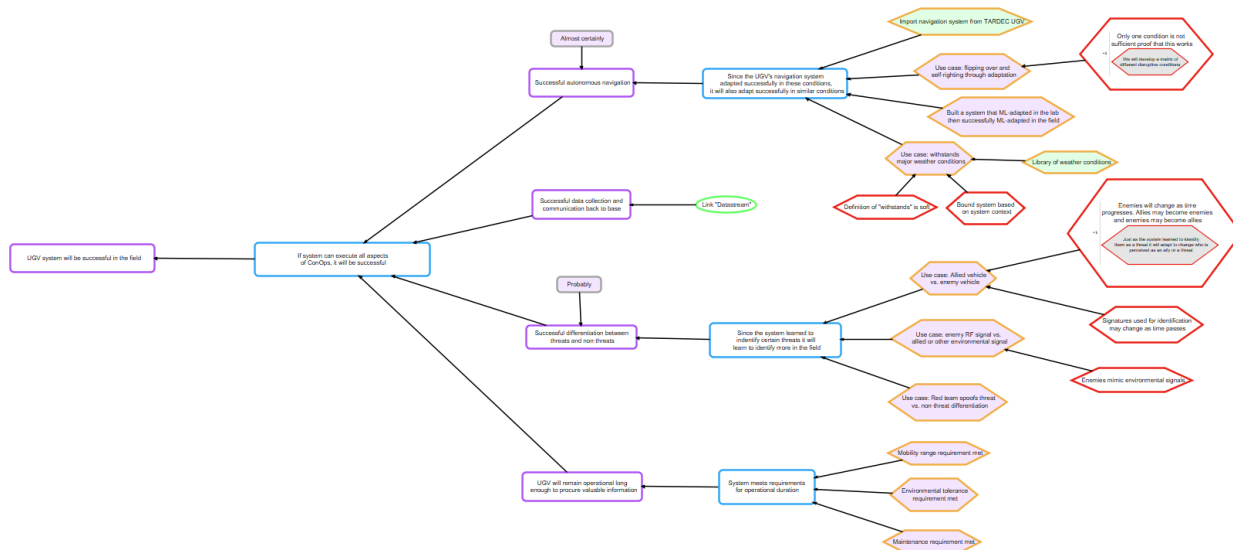


Figure 1-7. A second application.

In this graph, data are shaded green if the data is currently available, and purple if there is a plan to collect the data or run the test, but nothing is available yet. We also reserve yellow for a test or investigation that is being executed, but has not yet provided complete results. Because this

graph is quite large, and therefore the print is very small, Figures 1-8 through 1-11 show insets of elements of the graph.

In this example, we were able to refine the application of qualifiers and rebuttals (Figures 1-9 and 1-10). We added the notion that a warrant could be used to show how a set of claims might justify a more general claim (Figure 1-8). Also, in order to preserve a hierarchical structure that is necessary to describe a complex validation, we devised a link that can connect one graph to another (Figure 1-11).

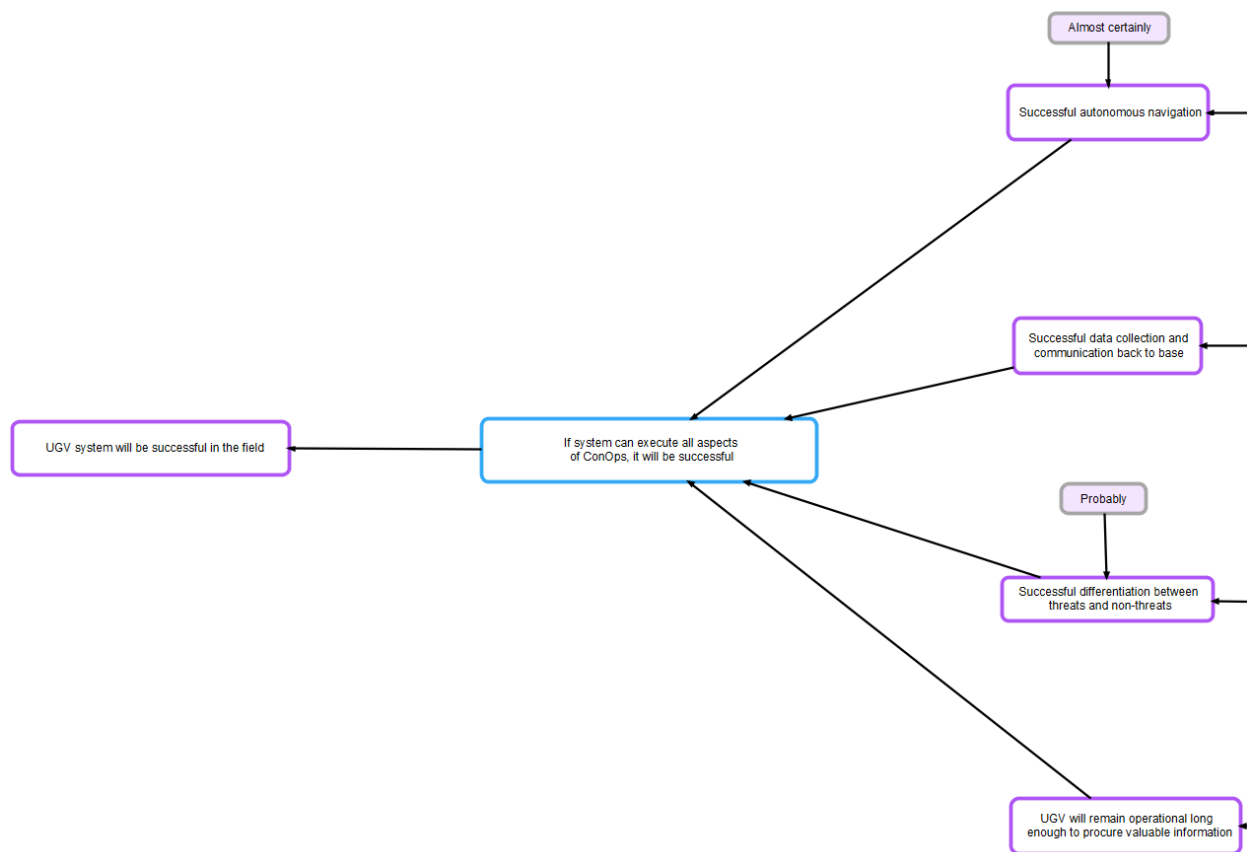


Figure 1-8. A general claim based on more specific claims.

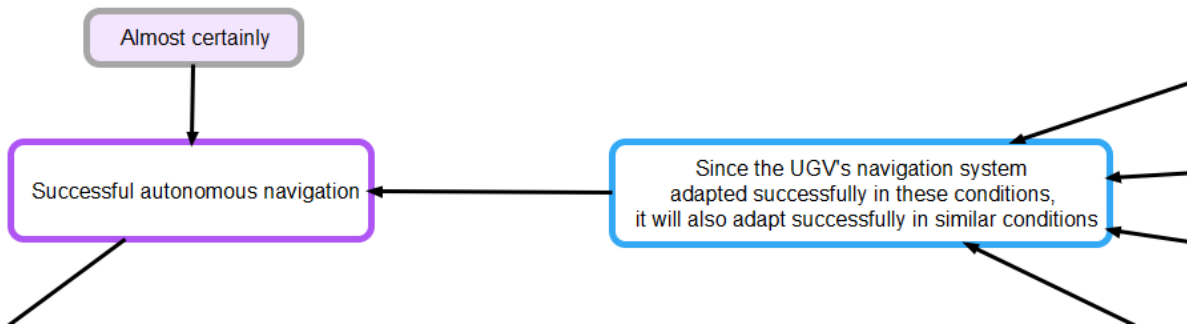


Figure 1-9. First sub-claim

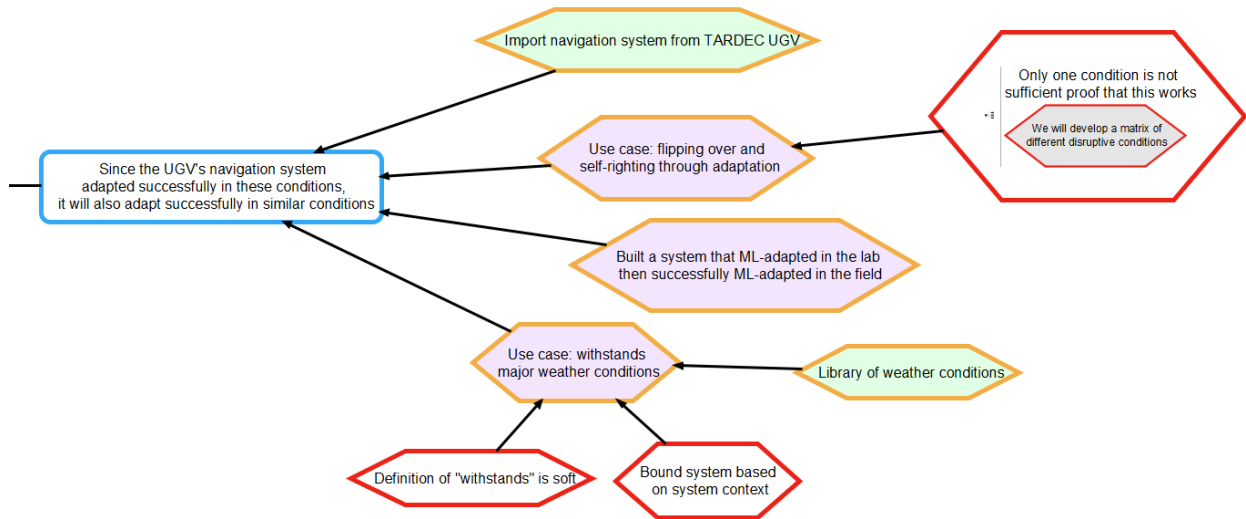


Figure 1-10. Data for first sub-claim

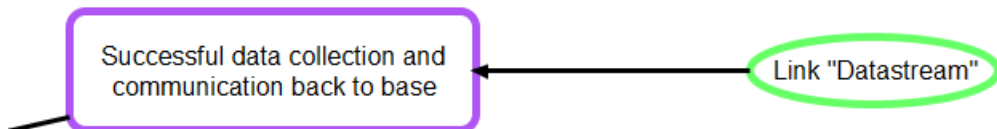


Figure 1-11. A link to warrant and data in another graph.

Rough feasibility of the method has been demonstrated, and much has been learned about applying it. A follow-on project could develop this validation method by applying it to actual systems and working with systems engineers to train them in using the method, then observing their performance to feedback improvements into the training. At this point, the method has shown that it might be quite effective, and no show-stoppers have appeared that would cast shadows on its feasibility for full-scale application to complex systems.

Advantages that we have observed for the Toulmin model in this feasibility study:

- Because the model is formal, it organizes a list of validation tests and show an explicit and reasoned connection between tests and the specific aspects and capabilities that are validated by the test.
- Warrants can capture the interaction between data from various tests and other sources. A surprisingly explicit and detailed result from one test may compensate for an unfortunately ambiguous result from another test.
- The tool used here is open source and Java-based. It can run on any browser. By enhancing the Java-code, we could greatly enhance the usability of the tool for creating and checking these validation models.

4.3.2 Phase 1 – Develop Methodological Foundations

Phase 1 will mature the explorations under the incubator award to develop a formal implementation of the Toulmin method applied to AI and LE systems. There are several tasks, some with varying degrees of parallelism, envisioned.

Task 1-1: Build background knowledge into an understanding of how to develop the warrant.

There are various ways to classify AI/ML algorithms. They may be grouped by learning style or by similarity in function, for example. How AI/ML algorithms may be classified with respect to how that classification impacts development of the warrant, its structure, and the validation process associated with that warrant must be defined and an approach selected. Additionally, the argument structure will need to be bounded to (a) reflect the AI aspects of system behavior in synergy with (b) the constraints associated with the physical system capabilities and envisioned operational contexts. The research needs to develop the foundations necessary to incorporate these aspects into argument structuring and development.

Task 1-2: Develop a defined process that will lead to development of the relevant argument structure for specific classes of AI and LE systems.

Based on the knowledge gleaned from Task 1-1, the research will develop and generate standard argument templates for a variety of regions on the AI/LE capability spectrum. These templates should also be able to characterize for the analyst/ decision maker where hardware in-the-loop (HWITL) and/or software in-the-loop (SWITL) testing will suffice to generate data to back a particular claim and when and where physical test and evaluation will be needed.

Task 1-3: Establish rationale for and calculation of confidence from the elements of the argument across classes of AI and LE systems. As the research begins to fill in the components of the Toulmin Argument Pattern and establish the correspondence of this method to the SE process, the calculation of confidence from the elements of the argument must be established. There are three primary dimensions associated with this challenge:

- i) *Inter-relationship between evidential bodies of data.* Many data generated to satisfy a claim may be related to other sets of data addressing a different claim. The research needs to specify a means to account for these potential relationships and define how the argument structure will reflect this aspect.
- ii) *Specificity bias.* In ML-based LE systems in particular, copious training may produce excellence in one operational context at the expense of producing extreme brittleness in another.

Task 1-4: Inform development via consideration of a real system and how the method would be embodied within a tool to facilitate its use. The research effort will work with an as-yet-to-be-determined system development program in the DoD to construct, in parallel but not in live use by the program so as not to hinder it, a validation argument as described above. Simultaneously, realization and repeatable use of the method requires a prototype tool to structure the warrant, claim, evidence, etc., and maintain the current state of confidence in the validation as it is developed over the course of the program. The research will conceptualize how the method and insights it brings could be captured in a tool/unifying framework for use by Systems Engineers. The work will describe, at a high level, how the method might be realized to help sort out relevant information, tie the right concepts together, and consequently draw from this information to structure arguments in line with the findings of the tasks above. Depending on the level of funding available and necessity to realize the approach, a prototype framework will either be expanded upon from existing tools used during the Incubator phase or developed separately. This task will explore using the tool to plan validation tests and to track the progress of validation.

Task 1-5: Conduct a usability study of the tool with working systems engineers. The tool design will be improved through a controlled study in a user-experience environment. We would like to conduct at least two iterations of study tool – improve tool – study tool – improve tool. The ARL-supported laboratory at UAH is well-designed for this type of study.

4.3.3 Phase 2 – Mature Methodology and Transition

Phase 2 will build from the research developed in Phase 1, focusing on transitioning the tool into practice with DoD engineering teams.

Because this research addresses a pressing need for which there is no current solution, we aim to transition the product to practice as soon as feasible so that continued maturation and best practices via application may be developed for the broader DoD/Federal community. Importantly, the methodology, despite the development of guidance frameworks for distinct types of AI and LE systems, will require an understanding of the argument process, its intent

and foundations, and how to best implement it effectively. Training will be an important consideration under this task.

Task 2-1: Develop a training program for validation planning and tracking using the Toulmin model. We will develop a DAU-style course to train advanced systems engineers in the method and use of the open-source tool. We will deliver the course once as a prototype test of the course, and then refine it. All course materials will be made available to DoD, but we also expect to integrate the course into the graduate curriculum at Georgia Tech and UAH.

Task 2-2: Transition the method into practice in at least one active DoD program. We plan to, in cooperation with our sponsor, select a DoD program to pilot live use of this method. We will work alongside engineers in the selected program to coach them in use of the model, and improve our tool and method based on their feedback. We will document the transition process, emphasizing effective approaches, to assist rolling out the method and tool more broadly.

4.3.4 Research Team Qualifications to Ensure Project Success

Together, the co-PIs bring expertise in the fundamental underpinnings of system engineering in the DoD Acquisition process (Paul Collopy, UAH) as well as trans-disciplinary engineering science, emphasizing hard-to-define analyses and design/assessment of fielded capabilities and JUON products (Valerie Sitterle, GTRI). Both PIs have strong backgrounds working with AI systems. GTRI excels at advancing systems engineering and analysis of complex defense systems using open-source technologies to build integrated toolsets based on modular open architecture software frameworks. The GTRI team includes researchers well-versed in complexity, MBSE standards, processes, and methods development, discrete simulation, and unique analytics approaches. UAH has built a world-class team of researchers in theoretical aspects of system engineering. UAH also has developed an extensive laboratory to study teaming of soldiers and autonomous systems, with an emphasis on usable human interfaces to AI applications. In addition, the team will include Daniel G. Shapiro (UAH), an expert in AI system development with over four decades of experience in machine learning, including directing a research institute at Stanford University focused on machine learning.

4.4 RELEVANCE OF THE RESEARCH

This effort addresses a fundamental research challenge in the *SERC thematic area of AI and Autonomy*, specifically *Lifecycle processes for AI systems, test, and evaluation for AI systems*.

By formally grounding a framework in systems science backed with an understanding of deployed capability ranges, this work will identify a validation path for AI and LE systems that will guide not only their design but also development of other methods and tools that will support the validation. Validation, in turn, is necessary to build assurance cases for AI and LE systems. This is a necessary enhancement of existing system validation, an understanding that will support a

rigorous development process for autonomous systems designed to exhibit novel behaviors in the field. Discussion of validating DoD AI and LE systems to date has relied on vague assertions such as “It will be tested in the field until we believe it will work.” Instead, a process is needed that produces a validation plan and an evaluation of the output of validation activities which yield varying levels of confidence supported by objective logic. That is precisely what this research seeks to develop. The research team expects this methodology to rapidly become a standard approach for AI systems, and eventually the preferred method for validation in all complex engineered systems.

4.5 RISKS, MITIGATIONS, AND PAYOFFS

Our central hypothesis is that an argument-based methodology can effectively produce a trusted path for AI and LE systems validation that will traceably identify data and test and evaluation activities necessary to support the validation and thereby support trusted behavior characterization for such systems.

Risk: The risk is that a high-level of abstraction when defining the components of the argument is not sufficient to accomplish this purpose. **Mitigation:** This is part of the fundamental research question for this effort and the hypothesis will be confirmed or refuted at the end of Phase 1.

Risk: AI and ML systems with different capabilities will require completely different argument structures, posing a challenge to repeatability. **Mitigation:** This is actually a key supposition of the research, and will also be confirmed or refuted. The team anticipates using a differing argument structure for the various levels of AI capability, the scalability and repeatability being achieved within-class for each.

Risk: The capabilities of the AI component of the system, being embodied in the argument methodology, are not separable from the physical capabilities of the engineering design. **Mitigation:** This is also a key part of the research that must be addressed. A system not designed for amphibious operation will not, for example, successfully engage in such even if the AI systems identifies that as the best course of action. The AI components in the argument must be bounded by feasibility with respect to the envisioned overall system design, and the delineation of what is encompassed on the argument and what is separate are a key outcome for this research effort.

Risk: The research needs a real problem to perform conceptual validation. **Mitigation:** GTRI, UAH, and the SERC have connections with various DoD groups and development efforts that will be approached for specifying a real system with real data and a targeted behavioral profile to assist with conceptual validation. This research will be performed in parallel with the real system development so as not to impact existing programmatic. Examples include the Advanced Combat Vehicle development at TARDEC.

Payoffs: The payoffs for this research, if successful, are tremendous. There is no current approach well-suited to validation for these types of systems. This research has the potential to become the standard approach for AI systems validation in support of DoD Acquisitions programs.

4.6 BUDGET

Although precise figures have yet to be calculated, the estimated budget is \$390K for Phase 1 over 18 months. Phase 2, also targeted for 18 months and similarly estimated at \$390K, will consist of targeted efforts for maturation and transition, driven by the results from Phase 1, to extend the methodology and associated framework into existing processes and application for wider use.

4.7 TIMELINE

Phase 0 of this effort was completed during the incubation project period and lasted 2 months from when funding was received until this report was produced. Phase 1, and Phase 2 are each projected to last 18 months. The total timeline outside of the pilot is projected at 3 years.

4.8 PROJECT EVALUATION AND MEASUREMENT

By the end of Phase 1, we expect to produce a minimal viable demonstration of our approach and a comprehensive description of our underlying conceptual advances for the SERC and DoD sponsors. By the end of Phase 2 – within 3 years - we anticipate significant demonstration of the methodology and tools in pilot applications at realistic scales.

Progress will be measured by annual technical review assessments made by key stakeholders across the SERC, USDR&E, and complimentary RT stakeholders across the SERC AI thematic area. These reviews will include a description of the conceptual advances and methods, processes, and tools that can support application to real problems as well as relevant demonstrations. The findings and results will be vetted with the stakeholder community for any necessary course-alteration and transition planning.

4.9 REFERENCES

Boehm, B. (2006). Some future trends and implications for systems and software engineering processes. *Systems Engineering*, 9(1), 1-19.

Button, R.W. (2017, Sept 7). Artificial Intelligence and the Military. RAND Commentary in *RealClearDefense*. [HTTPS://WWW.RAND.ORG/BLOG/2017/09/ARTIFICIAL-INTELLIGENCE-AND-THE-MILITARY.HTML](https://www.rand.org/blog/2017/09/artificial-intelligence-and-the-military.html)

Caminada, M. (2006). Semi-stable semantics. In P. E. Dunne & T. J. M. Bench-Capon (Eds.), *Computational models of argument*. Proceedings of COMMA 2006, September 11–12, 2006,

Liverpool, UK (Frontiers in artificial intelligence and applications, Vol. 144). Amsterdam: IOS Press.

Chesnevar, C., Modgil, S., Rahwan, I., Reed, C., Simari, G., South, M., Vreeswijk, G., and Willmott, S. (2006) Towards an argument interchange format. *The Knowledge Engineering Review* 21(4) 293-316.

Dahm, W. J. (2010). *Technology Horizons a Vision for Air Force Science & Technology During 2010-2030*. USAF HQ, Arlington, VA.

DARPA Prize Authority (2006, Mar). DARPA Prize Authority Report to Congress, March 2006, Defense Advanced Research Projects Agency.

DARPA Urban Challenge (2007). <https://www.grandchallenge.org> (Accessed 1 December 2019).

DARPA (2014). The DARPA Grand Challenge: Ten Years Later. News and Events, Defense Advanced Research Projects Agency, <https://www.darpa.mil/news-events/2014-03-13> (Accessed 1 December 2019).

DARPA (2016, Aug 10). Explainable Artificial Intelligence (XAI) DARPA-BAA-16-53 Broad Agency Announcement (PM: David Gunning).

DARPA (2017, Aug 15). Assured Autonomy. I2O Program Description (PM: Sandeep Neema). [HTTPS://WWW.DARPA.MIL/PROGRAM/ASSURED-AUTONOMY](https://www.darpa.mil/program/assured-autonomy) (Accessed 16 April 2019).

Dung, P. M. (1995). On the acceptability of arguments and its fundamental role in non-monotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77, 321–357.

Erduran S., Simon S. and Osborne J., (2004), TAPping into argumentation: developments in the application of Toulmin's argument pattern for studying science discourse, *Sci. Educ.*, 88(6), 915–933.

Felder W. and Collopy P. (2012). The elephant in the mist: what we don't know about the design, development, test and management of complex systems, *Journal of Aerospace Operations*, 1(4), 317–327, DOI10.3233, IOS Press.

Heule, M. J., & Kullmann, O. (2017). The science of brute force. *Commun. ACM*, 60(8), 70-79.

Jakobovits, H., & Vermeir, D. (1999). Robust semantics for argumentation frameworks. *Journal of Logic and Computation*, 9(2), 215–261.

Li, L., Lin, Y. L., Zheng, N. N., Wang, F. Y., Liu, Y., Cao, D., ... & Huang, W. L. (2018). Artificial intelligence test: a case study of intelligent vehicles. *Artificial Intelligence Review*, 50(3), 441-465.

Prakken, H. (2010). An abstract framework for argumentation with structured arguments. *Argument and Computation*, 1, 93–124.

Roff, H.M. and Danks, D. (2018). “Trust but Verify”: The difficulty of trusting autonomous weapons systems. *Journal of Military Ethics*, 17(1), 2-20.

Stanford University (2019). Propositional Satisfiability, Stanford Introduction to Logic: An Online Course on Symbolic Logic, <http://intrologic.stanford.edu/extras/satisfiability.html> (Accessed 2 December 2019).

Stevens, J. S. (2017, July). Warranting System Validity Through a Holistic Validation Framework: A Research Agenda. In *INCOSE International Symposium* (Vol. 27, No. 1, pp. 654-671).

Toulmin, S. E. (2003) *The Uses of Argument*. Cambridge University Press; Updated edition. ISBN-13: 978-0521534833.

Toulmin, S. E., & Toulmin, S. (1992). *Cosmopolis: The hidden agenda of modernity*. University of Chicago Press.

F.H. Van Eemeren et al., 2014. *Handbook of Argumentation Theory*, Springer. DOI 10.1007/ 978-90-481-9473-5

Verma D. (2018). *Systems Engineering Research Needs and Workforce Development Assessment*. Technical Report SERC-2018-TR-102, January 31, 2018. Also available through DTIC.

Wilson, B. (2018). *Interfacing Force-on-Force and Communications Models: MANA and JNE* (No. TL-201-A). RAND Corp Santa Monica United States.

Zhang, X., Clark, M., Rattan, K., & Muse, J. (2015, April). Controller verification in adaptive learning systems towards trusted autonomy. In *Proceedings of the ACM/IEEE Sixth International Conference on Cyber-Physical Systems* (pp. 31-40). ACM.

5. FOSTERING HUMAN LEARNING FROM COGNITIVE ASSISTANTS FOR DESIGN SPACE EXPLORATION – DR. DANIEL SELVA, TEXAS A&M UNIVERSITY

5.1 EXECUTIVE SUMMARY

The problem we are trying to solve is how to develop cognitive assistants for design space exploration that foster—as opposed to hinder—human learning. The objective of this work is to demonstrate a new approach to foster learning in cognitive assistants based on a dedicated agent called a Teacher. To do that, we first define human learning in this context and propose ways to measure it. Then, borrowing from education theory and the intelligent tutoring systems literature, we implement a variety of strategies to foster learning, based on just-in-time notifications with questions and information designed to help the user gain new insights about key sensitivities and features in the problem. We also look at how to adapt to individual preferences and differences such as cognitive style. We validate the effectiveness of the Teacher agent in an experiment with human subjects.

5.2 BACKGROUND AND MOTIVATION

Cognitive Assistants (CA) such as Siri or Alexa are becoming ubiquitous in our homes as we use them for mundane tasks in our daily lives. We argue that the time is ripe for adopting this Artificial Intelligence (AI) technology in the workplace, and specifically to support systems engineering tasks.

The motivation is two-fold: on one hand we have a technology push, as recent advances in machine learning in general and natural language processing in particular have increased the performance of this kind of agents to the point where the experience of interacting with them is no longer dominated by the low quality of the natural language interface. On the other hand, we have a societal push, since we as individuals and as a society have become relatively comfortable interacting with such AI agents.

In our lab, we have been researching for several years how to use CA to support systems engineering. This proposal focuses on the specific task of Design Space Exploration (DSE), a.k.a. tradespace exploration [1], or design by shopping [2]. DSE is an iterative, information-gaining process done early on in the system formulation phase, by which systems engineers use computational tools to generate and evaluate a large number of alternative system designs on a number of figures of merit, such as performance, lifecycle cost, various types of risk, flexibility or other lifecycle properties (–ilities). The process starts by defining a design in terms of a number of design decisions, which together with their allowed values define the *solution space* or *design space* to be explored. Then, an

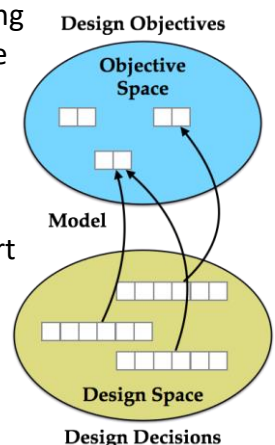


Figure 16: DSE uses one or more models to map design decisions (Design Space) to figures of merit (Objective Space)

objective space is defined in terms of a number of figures of merit. Specifically, DSE requires two main types of tools: *models* (e.g., simulations, cost models) to calculate the figures of merit as a function of the design decisions (i.e., map the design space to the objective space, cf Figure 1), and *search or optimization algorithms* (e.g., full factorial enumeration, genetic algorithms) to generate alternative designs. The output of DSE is typically a large dataset of evaluated alternatives. At this point, one can use visual and data analytics tools to make sense of the data and draw conclusions [3]–[6].

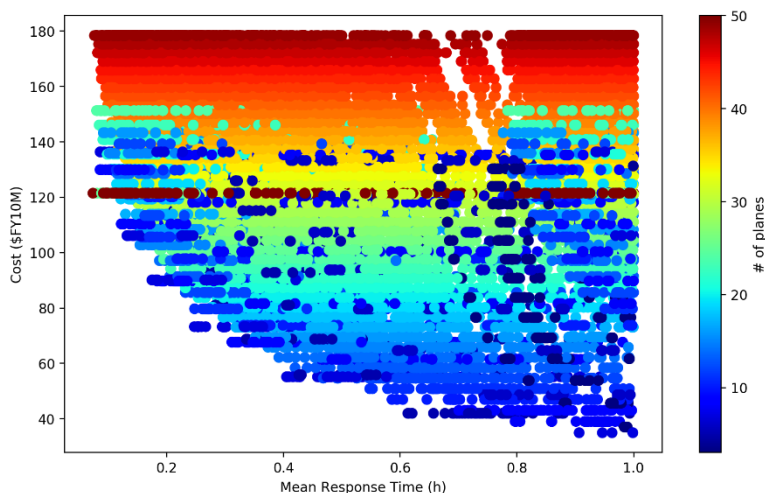


Figure 17: A scatter plot representative of typical outputs of DSE processes. Each dot is an architecture in objective space.

While DSE is gaining popularity as a means to learn useful information about the design problem early on and facilitate design decision making, there are several challenges that reduce its utility in real-world applications. First, there is high risk of information overload. Consider a typical example output of a DSE process in the form of the scatter plot of Figure 2. The figure contains hundreds of thousands of dots, each of which represents a different design in objective space, in this case of a constellation of weather satellites. The designs are colored according to an important design variable, namely the number of planes. We observe an underlying smooth pattern with what appears to be a noisier pattern on

top, particularly close to the Pareto frontier (the set of designs for which no other design can be found that is better across all metrics). Even if we restrict ourselves to the Pareto-optimal designs, it is easy to feel overwhelmed by the sheer number of designs to consider. Currently, to make sense of the results, we rely on extensive manual processes which include generating various plots or tables to look at specific subsets of the dataset and consulting various databases to find relevant information among other things. Moreover, more often than not, the models used are biased in a way that substantially affects the results that we are observing. In fact, optimization algorithms are notoriously good at finding flaws in our models. So, in reality, we are simultaneously doing model validation and design decision making, which adds a layer of complexity to the problem.

We argue that CA can help alleviate some of these challenges by providing the right information to the system engineer at the right time. In fact, in previous research funded by NASA and NSF, we have developed a CA called Daphne that is meant to support systems engineers in the early formulation of Earth observation missions [7]–[9].

In addition to providing models for estimating the performance and cost of such systems, state-of-the-art optimization algorithms and visual and data analytics tools, Daphne provides a natural language interface for question answering. For example, Daphne can answer questions such as

“What do you think of this design?” or “Has this ever been done before?” and provide some explanations for its recommendations as well as just-in-time pieces of relevant information as the user explores the design space (e.g., “I noticed a pattern among the best designs in this region”). The vision for Daphne is that it could become a true peer or companion for the system engineer during the early phases of mission design.

Like most CA, Daphne is essentially a question answering system with a natural language interface. The architecture of Daphne is shown on Figure 4.

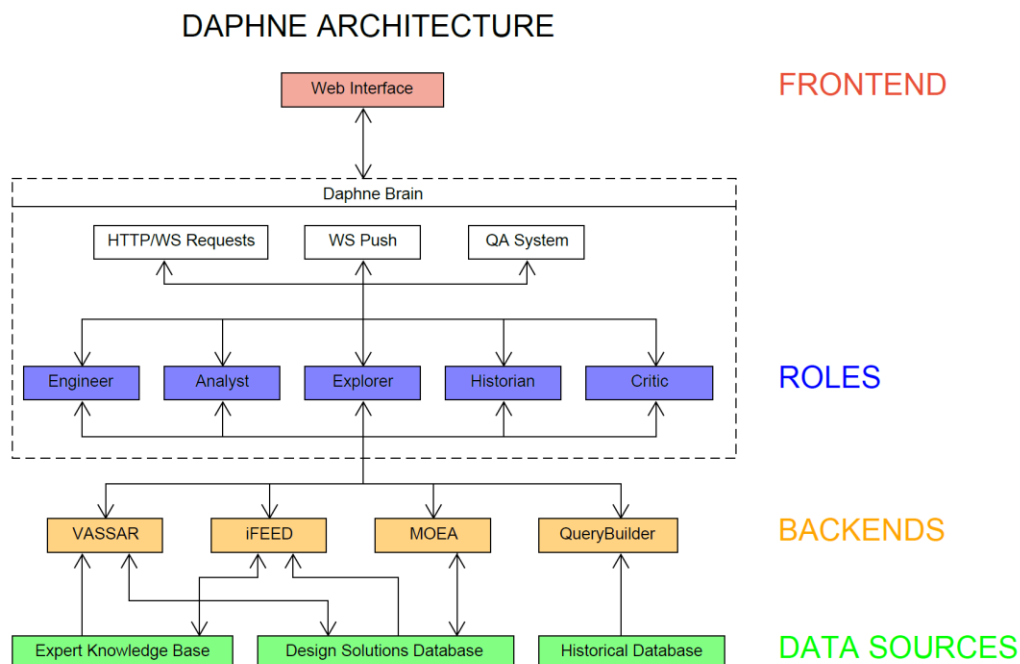


Figure 4: Architecture of the Daphne CA for architecting Earth observation missions

Daphne has five main components: 1) the front-end, which is a web-based graphical user interface (see Figure 5); 2) what we call the Daphne brain, which includes the question answering system and all the web-based communication infrastructure (e.g., http and websockets); 3) several *roles* or *skills*, which are agents that specialize in answering specific kinds of questions (e.g., engineering questions, questions about past missions); 4) the back-end services that those roles use to answer questions (e.g., VASSAR is a spacecraft design tool [10]); 5) the data sources used by the back-end services (historical mission database, expert knowledge base, design solutions database).



Figure 5: Snapshot of part of Daphne's web-based user interface (note the chat box on the right where the dialogue occurs).

In past work, we have shown that Daphne does indeed increase performance in DSE [7]. To do this, we conducted a study with 9 JPL engineers in which participants solved two similar DSE tasks: one with all the Daphne functionality (Daphne-VA in Figure 6), and one with only the traditional visual and data analytics capabilities, without the natural language interface (Daphne-DM in Figure 6). We then measured their performance in terms of the quality and diversity of all designs generated by the user, as measured by HyperVolume (HV) of the set of designs, which is a metric typically used to assess performance in multi-objective optimization problems (specifically, the HV of the union of the regions of the objective space dominated by each design). Figure 6 (left) shows that the people using the Daphne CA functionality (Daphne-VA) achieved (statistically significantly) higher HV on average than people using only the traditional functionality (Daphne-DM). Moreover, we also showed that the HV of the synergistic human-Daphne combination was higher than that of either the human alone or Daphne alone, whether the comparison is done on an iso-NFE (number of function evaluations) basis or an iso-time basis (cf Figure 6 center and right respectively).

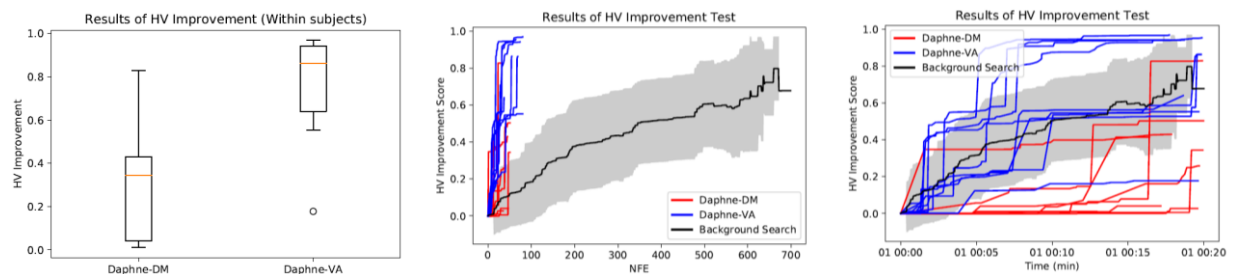


Figure 6: Results of the human subjects study conducted at JPL for the hypervolume (HV) measure. Left: HV improvement when using full Daphne functionality (Daphne-VA) vs traditional functionality only (Daphne-DM). Center: HV improvement as a function of NFE for each subject + Daphne and for Daphne alone (black line and

gray confidence interval). Note that Daphne tries many more designs than humans. Right: same comparison but on an iso-time basis. Note that all humans+Daphne-VA trials are at least comparable if not superior to Daphne alone.

However, in this study we also observed a worrying trend (though not statistically significant with the small sample size we got) suggesting that Daphne may actually *decrease* human learning. We mentioned earlier that DSE is above all an information-gaining process, so we argue that measuring how much humans learn in the process should be the goal – or at least is a factor important enough to consider. Of course, human learning is a challenging thing to define, let alone measure. We have ongoing (unpublished) research in the lab on various measures of human learning in DSE and how they are correlated, but for the purposes of this experiment, we used a simple test-based approach. Specifically, participants were asked 10 multiple choice questions on the DSE task, such as “Which of the two designs below do you think is on the Pareto front”, or “Do you think most (at least 70%) of designs on the Pareto front have this feature (e.g., fly the radar and the radiometer together)?”. The questions were carefully designed to make them at the right level of difficulty. The test score (number of questions answered correctly) was taken as a proxy for how much they learned in the task. Figure 7 (left) shows the test scores for each subject in the two conditions. Figure 7 (right) shows the same results in a boxplot. While the results are not statistically significant due to the high variance observed, it is easy to see that most lines have a negative slope on the left chart, and the Daphne-DM boxplot is mostly above the Daphne-VA boxplot on the right chart.

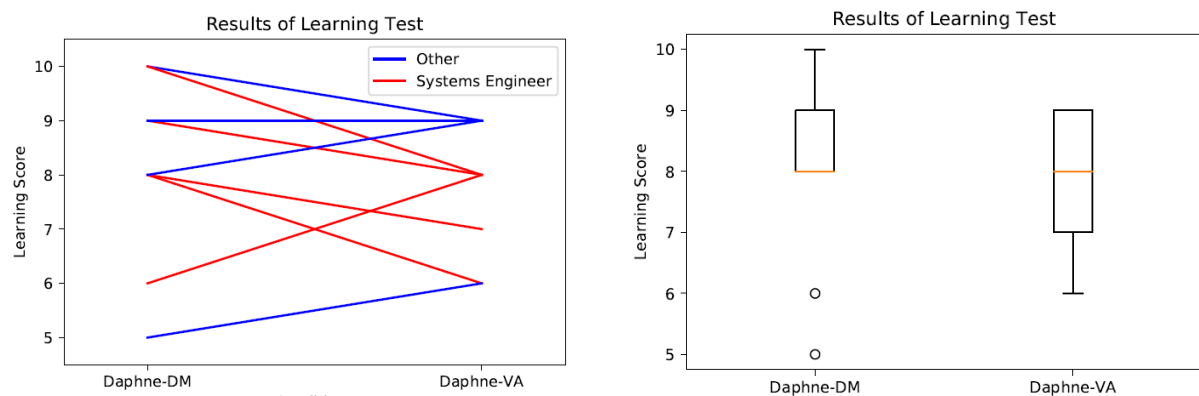


Figure 7: Results of the study for the human learning measure. Left: Test score for each subject in each condition. Right: Boxplots of test scores for each condition. While results are not statistically significant, they show a concerning trend.

As we have learned since then, this measure suffers from a number of limitations such as low precision and high variance. Still, we have observed similar results in other similar studies we have run (though with student populations [8]), so the concern remains.

5.3 RESEARCH OBJECTIVES

The goal of this project is to develop approaches to foster —as opposed to decrease—human learning in human-CA collaborative DSE and demonstrate their effectiveness. Specifically, we have the following research objectives:

1. Measuring learning
 - a. Develop measures of human learning that allow us to monitor learning in multiple areas of DSE over time with high consistency.
2. Fostering learning
 - a. Develop a number of interventions (e.g., notifications containing summaries, questions, short insights and explanations) by Daphne with the purpose of fostering learning in those areas.
 - b. Develop strategies to select interventions that are more likely to foster learning in areas where the human is showing weakness.
3. Adapting to individual differences
 - a. Obtain feedback from the user about the perceived effectiveness of those interventions.
 - b. Demonstrate how we can adapt to user preferences and commands regarding those interventions.
4. Validation
 - a. Develop a prototype of Daphne that incorporates a new role called the “Teacher” implementing all the functionality described in the previous three objectives.
 - b. Measure the impact of the Teacher agent on human learning in a controlled experiment with human subjects.

5.4 TECHNICAL APPROACH

Defining learning

The first step in measuring anything is defining it. Learning is gaining knowledge.

But what knowledge do systems engineers learn when interacting with DSE tools? There are several dimensions of knowledge. On a conceptual dimension, we identify several areas of relevant knowledge in DSE: 1) sensitivities of figures of merit to design decisions; 2) couplings between design decisions (through constraints or through figures of merit); 3) driving design features, i.e., combinations of values for design decisions that consistently lead to a specific area of the objective space (e.g., the Pareto front, or a certain level of performance); 4) what-if questions related to the various model assumptions.

Importantly, we care about these areas of knowledge mostly through their application to creating or selecting better designs. In other words, we make the assumption that engineers who know more about sensitivities and driving features will be able to *apply* that knowledge and come up

with better designs. However, this is a strong assumption that should be tested (note that our preliminary work suggests that this is not true, although those results may be invalid due to the high variance observed in the learning measure).

The fact that learning should be measured not just in terms of factual knowledge but also on its application to solve higher level problems (e.g., creating new designs) is well known in the education literature. Bloom's taxonomy, for example, provides a hierarchy of educational objectives that span a wide range of skills and abilities, from simple to complex and from concrete to abstract [11], [12]. Inspired by Bloom's taxonomy, we could argue that our test was mostly focusing on

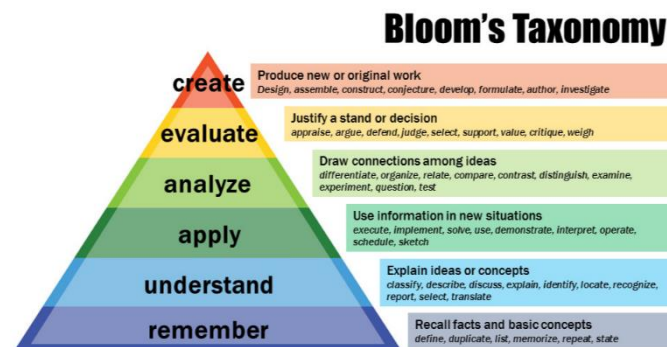


Figure 22: Bloom's taxonomy from the education literature

the lower levels of the pyramid plus the very top (create new designs), but we should at least consider learning across the entire range of Bloom's taxonomy.

Measuring learning

We will develop multiple complementary strategies to measure learning in DSE, including:

- A more comprehensive test-based approach with more questions covering a wider range of Bloom's taxonomy, from remembering basic information about the design problem (e.g. achievable range of performance) to analyzing or describing the structure of the tradespace in terms of a few clusters or families, or to evaluating or predicting the strengths, weaknesses and ultimately performance of a design.
- An approach based on having users explicitly represent their new knowledge in the form of a knowledge graph, so we can use simple graph-theoretic concepts to measure learning as the distance between two graphs (before and after the task). This assumes that people who learn more will make more changes (i.e., add new concepts and relations) to their initial knowledge graph before the task.
- A subjective measure of learning based on a simple survey including Likert items related to how much they think they learned.

We will then look at the statistical correlations and consistency between all these different measures as well as with the design synthesis measure (i.e., hypervolume).

All the aforementioned approaches have an important limitation, namely that they assume that Daphne knows the true answer to all these questions. But this is of course not the case, due to at least two factors: model errors, and insufficient sampling of the design space. In other words, as the search through the design space progresses, the true answers to the questions may evolve. For example, the driving features that are common to architectures on the Pareto front may evolve as the optimization algorithm improves the Pareto frontier. The intelligent tutoring systems community has tackled this problem, specifically in the field of test theory without an answer key [13]. The basic idea is to jointly estimate the skill of the subject in a particular area

and the actual answers to the questions in the test, simultaneously, using for example Markov chains [14] or hierarchical Bayesian models [15]. We will implement such Bayesian models to the context of DSE.

Fostering learning

We will develop a set of notifications (interventions or learning opportunities in education theory) in the Daphne CA targeting a wide range of skills and abilities from Bloom's taxonomy. In addition to the comprehensive test-based approach described above, we will also develop a Diversifier role. The goal of the Diversifier is to foster exploration of the design space. Diversity is to some extent implicitly included in the hypervolume metric. However, the assumption is that explicitly fostering exploration of a diverse set of designs will lead to more learning. We believe this may be especially important when the dataset is small and an optimization algorithm is being used to increase it, since optimization algorithms will tend to introduce sampling bias (towards features present in better architectures) and this may lead to inaccurate estimates (by humans and computers alike) of sensitivities, for example.

The Diversifier considers information from both the objective space (diversity in performance) and the decision space (diversity in implementations). The objective space diversifier uses a diversity metric from the multiobjective optimization literature (crowding distance [16]) to measure the local density of solutions in a region of the objective space. To foster diversity in the decision space we borrow from the theory of design of experiments and look at how many times a combination values for a set of decisions has been sampled by the user. In both cases, the idea is to keep track of unexplored areas and propose the user to explore some of those. The agent will have a reactive aspect that answers questions from the user and a proactive aspect that can push notifications to the user through Web Sockets. In addition to showing notifications, the Diversifier will also be able to search for designs from unexplored areas using constraint handling mechanisms such as the Lagrange penalty method.

In addition to the questions and the Diversifier, the Teacher agent will also show notifications including short summaries of the structure of the design space or other relevant pieces of information concerning the areas of knowledge identified in the previous section (i.e., sensitivities, couplings, driving features, etc.)

Adapting to individual differences

It is well known that human-computer interaction is largely driven by individual differences [17]. Hence, in fostering learning, it is important to know those individual differences and be able to adapt to them. While some of that will be done implicitly by our Bayesian approach to measure learning, some of it can also be done more explicitly by taking into account cognitive style differences measured by well-known and validated standard surveys and inventories [18]–[20].

We will thus develop the code necessary to accept feedback from the user after showing them a question or notification and also to accept information about individual user differences in the form of standard cognitive style surveys, and incorporate that information in the selection of the next question or notification to show to the user. For example, our pilot studies have shown that

users want to be able to select the frequency with which they receive various types of notifications from the Teacher agent.

Validation

The approach to validate the effectiveness of the Teacher agent will be to conduct a pilot study with human subjects. Specifically, we will run a counterbalanced within-subjects experiment similar to the one described in the background section, but where the two conditions will be Daphne with Teacher vs Daphne without Teacher. Each participant will work on two similar DSE tasks, one in each condition. We will measure HV as in the original study, plus the measures of human learning described earlier in this section. We hypothesize that on average, participants will learn more when using the Teacher agent. It will also be interesting to see if improved learning leads to improve design performance as measured by the hypervolume or not.

5.5 PRELIMINARY WORK PERFORMED IN INCUBATOR PHASE

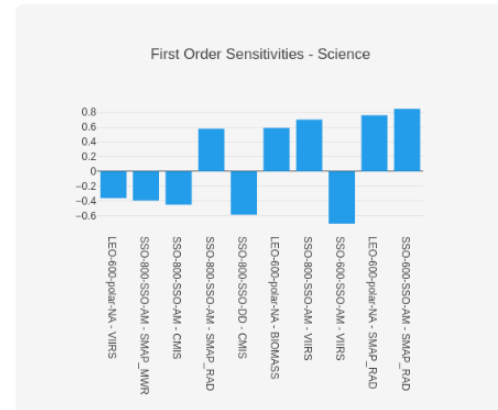
During the Incubator phase of this project, we have developed the first prototype of the Daphne *Teacher* agent including the following functionalities:

- Show relevant information to the user about key sensitivities of the design problem. This is measured by periodically calculating in the background Sobol's first-order sensitivity indexes for all design decisions and all figures of merit. For example, the Teacher agent may alert the user that the decision of whether or not we fly a radar altimeter and a lidar in the same spacecraft is driving most of the variability in cost. A snapshot of what this looks like is shown on Figure 8.
- Show a suggestion to the user when there is a region of the objective space that is underexplored. This is measured by means of average crowding distance in that region. For example if the low-cost region of the objective space has not been explored at all, Daphne will suggest to create some designs with low cost to explore that area. The corresponding plot is shown in Figure 10. The next step is to support the user in this goal by performing some constrained search using level-set genetic algorithms.

Teacher: I have found information about sensitive design decisions, would you like to learn more?

Yes

Teacher: Ok, here is what I found on first order sensitivities for the Science objective



Ask a question / Give a command / Speak it out!

Send

Figure 23: Example dialogue with Daphne Teacher agent

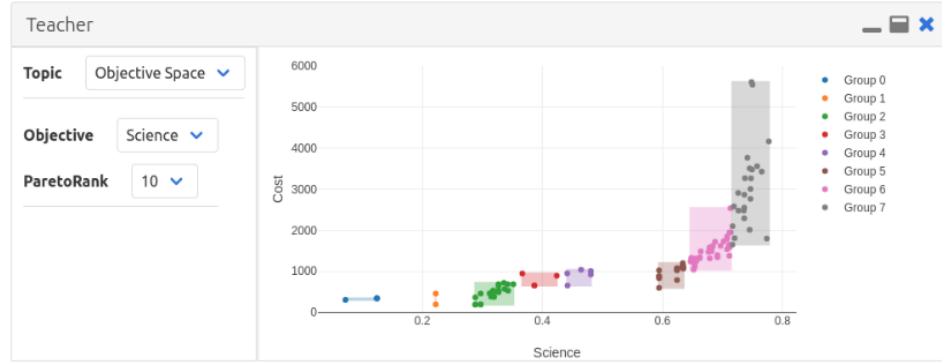


Figure 10: Example plot generated by Teacher to identify and explain underexplored region of objective space

- Show a suggestion to the user when there is a region of the design space that is underexplored. This is measured by the number of times that a certain combination of decision values has been sampled in the design space. For example, if a certain combination of instruments and orbit has never been tried, Daphne will suggest that the user explores some designs with that feature. The next step is to support the user in this goal by performing some constrained search with standard constraint handling methods.
- Ask questions to the user to support learning in an explicit fashion. Different types of questions have been implemented, spanning a wide range of task types as described in Bloom’s taxonomy:
 - **Factual information:** “Which of these decision variables has a larger impact on cost?”
 - **Prediction:** For example: “Is the following design going to be on the Pareto front?”, or “Which of these two features covers a larger fraction of Pareto-optimal designs?”
 - **Synthesis:** For example, “Create a design that is as close to the Pareto front as you can”

In addition, we have planned and are currently running a pilot experiment to measure the efficacy of the Teacher agent. We have the following hypotheses: H1) Test subjects using Daphne along with the Teacher Agent will produce a better and more diverse set of architectures (as measured by the hyper-volume) than those that use Daphne without the Teacher Agent. H2) Users will learn more about the design problem (as measured by the test) when using the Teacher Agent with Daphne than users that use Daphne without the Teacher Agent. The details of the experiment are as follows.

Population: Nine students at Texas A&M participated in this pilot study (6 male, 3 female). Three of these students were undergraduates and the other six were graduate students, all of which are pursuing Aerospace Engineering degrees.

Procedure: Each was tasked with designing a DSM (Distributed Spacecraft Mission) for monitoring Soil Moisture. The task goal was to find the best possible set of designs in terms of science and cost (maximizing the hyper-volume of the set of designs in the objective space). This is the same task that was used in the JPL study.

Measures: To determine how much each candidate learned about the problem at hand, a test of 40 questions was administered at the end with questions on the topics mentioned above (i.e., sensitivities, good designs, driving features, etc.) The number of correct answers is the proxy for learning. Other measures of learning that are being developed are not included as part of this pilot experiment.

Experimental design: Counter-balanced within-subjects experiment. Each task and test was administered twice, once using only Daphne and once using Daphne with the Teacher Agent.

Preliminary results: The preliminary results are shown on Figure 11, which shows the learning test scores for design prediction questions.

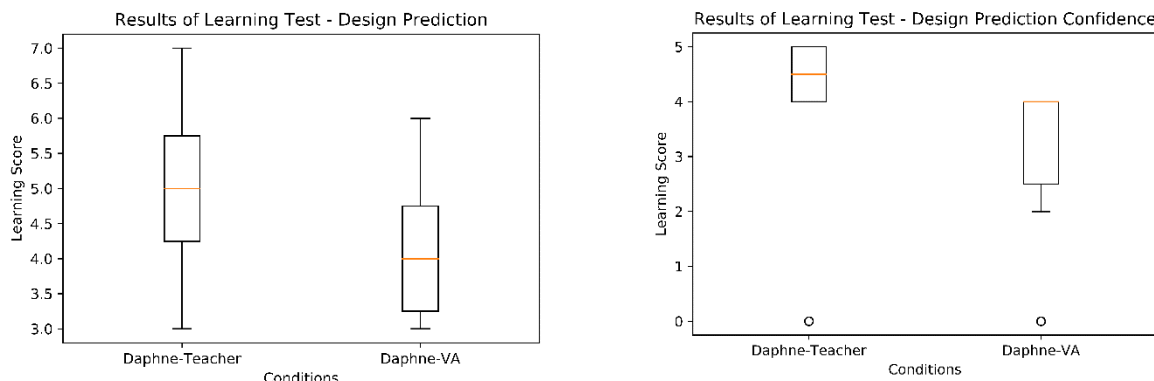


Figure 11: Preliminary results for the pilot experiment. Boxplots for test scores with and without the Teacher agent. Left: all results. Right: results only taking into account answers were user had high confidence.

Discussion: Positive trends are observed in the design prediction questions, but not on feature prediction or sensitivity questions. While design prediction questions are arguably the closest ones to the task at hand, we still want to understand why the first prototype of the Teacher agent is not improving learning on the other types of questions. We hypothesize that some of this is due to the participants not fully understanding the concepts of global sensitivity or driving feature in the short time allotted for the experiment, so the actual experiment will be longer and ideally with expert participants.

5.6 PROPOSED RESEARCH

Given the preliminary work performed during the Incubator phase, and the research objectives we set forth to achieve, we propose a research plan consisting of the following tasks:

Year 1

Task 1: Continue characterization of different measures of learning in DSE.

Task 2: Develop Hierarchical Bayesian model to measure learning while simultaneously estimating the true answers to Daphne's questions.

Task 3: Refine implementation of the Teacher agent and add new interventions, including short summaries of the dataset.

Task 4: Develop strategies to select interventions that are more likely to foster learning in areas where the human is showing weakness.

Task 5: Design and conduct experiment to validate the effectiveness of the Teacher agent with expert subjects. The PI will work with the customer to identify potential participants in the experiment.

Year 2

Task 6: Define a new task that is more relevant for customer needs, if applicable.

Task 7: Continue refinement of the Teacher agent and add new interventions, including short summaries of the dataset.

Task 8: Develop ability to adapt to individual differences, including both information on the user cognitive style gathered prior to start using the tool as well as preferences indicated during usage of the tool.

Task 9: Design and conduct a second experiment to validate the effectiveness of the updated Teacher agent in the more relevant task. The PI will work with the customer to identify potential participants in the experiment.

5.7 MANAGEMENT PLAN

The research team consists of Prof. Selva (PI), Mr. Gabriel Apaza (graduate student), and a second graduate student to be hired. We have identified a very promising candidate who is currently an undergraduate student at Texas A&M with a 4.0/4.0 GPA and research and industry experience in space mission design and design optimization. We will also engage one or two undergraduate researchers in this project. We will meet every other week as a team and every other week one-on-one with each team member.

Budget: The period of performance for the proposed work is 2 years: 1 year baseline period with an option of a second year to add more functionality and demonstrate the Teacher agent in a different problem more relevant to the customer's needs, if applicable. The total budget requested is \$331,673 (\$162,879 in year 1, \$168,794 in year 2) to cover 1 month/yr salary support (+ fringe and insurance) for the PI, 2 graduate students (tuition, stipend, fringe, insurance), \$4,000/y for an undergraduate researcher, \$6,384/yr for travel to progress meetings at the sponsor location and 1 domestic conference, \$2,000/yr for computing services, \$2,000/yr for publication costs and \$1,200/yr for conference fees, with a total IDC of 50% the first year and 51.5% the second year.

5.8 REFERENCES

- [1] A. M. Ross and D. E. Hastings, "The tradespace exploration paradigm," *INCOSE Int. Symp. 2005*, p. 13, 2005.
- [2] R. Balling, "Design by Shopping: A New Paradigm?," in *Proc. of the Third World Congress of Structural And Multidisciplinary Optimization*, 1999.
- [3] M. J. Woodruff, P. M. Reed, and T. W. Simpson, "Many objective visual analytics: rethinking the design of complex engineered systems," *Struct. Multidiscip. Optim.*, vol. 48, no. 1, pp. 201–219, 2013.
- [4] X. Yan, M. Qiao, T. Simpson, J. Li, and X. Zhang, "Work-Centered Visual Analytics to Support Multidisciplinary Design Analysis and Optimization," in *12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSM*, 2012.
- [5] N. Knerr and D. Selva, "Cityplot: Visualization of High-Dimensional Design Spaces With Multiple Criteria," *J. Mech. Des.*, vol. 138, no. 9, p. 091403, 2016.
- [6] H. Bang and D. Selva, "IFEED: Interactive feature extraction for engineering design," in *Proceedings of the ASME Design Engineering Technical Conference*, 2016, vol. 7, pp. 1–11.
- [7] A. Viros and D. Selva, "Daphne : A Virtual Assistant for Designing Earth Observation Distributed Spacecraft Missions," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, pp. 1–19.
- [8] A. Viros Martin and D. Selva, "From Design Assistants to Design Peers: Turning Daphne into an AI Companion for Mission Designers," in *AIAA Information Systems-AIAA Infotech at Aerospace, 2019*, 2019.
- [9] H. Bang, A. Viros, A. Prat, and D. Selva, "Daphne : An Intelligent Assistant for Architecting Earth Observing Satellite Systems," in *AIAA SciTech 2018*, 2018.
- [10] D. Selva and E. F. Crawley, "VASSAR: Value assessment of system architectures using rules," in *Aerospace Conference, 2013 IEEE*, 2013.
- [11] B. S. Bloom, C. of C. and U. Examiners, and University, *Taxonomy of educational objectives*, vol. 1. David McKay New York, 1956.
- [12] L. W. Anderson, D. R. Krathwohl, and B. S. Bloom, *A taxonomy for learning, teaching, and assessing: A revision of Bloom's taxonomy of educational objectives*. Allyn & Bacon, 2001.
- [13] W. H. Batchelder and A. K. Romney, "Test theory without an answer key," *Psychometrika*, vol. 53, no. 1, pp. 71–92, 1988.
- [14] G. Karabatsos and W. H. Batchelder, "Markov chain estimation for test theory without an answer key," *Psychometrika*, vol. 68, no. 3, pp. 373–389, 2003.
- [15] Z. Oravecz, R. Anders, and W. H. Batchelder, "Hierarchical Bayesian Modeling for Test Theory Without an Answer Key," *Psychometrika*, vol. 80, no. 2, pp. 341–364, 2015.
- [16] K. Deb, M. Mohan, and S. Mishra, "A fast multi-objective evolutionary algorithm for finding well-spread pareto-optimal solutions," 2003.
- [17] J. Y. C. Chen and M. J. Barnes, "Human-Agent Teaming for Multirobot Control: A Review of Human Factors Issues," *IEEE Trans. Human-Machine Syst.*, vol. 44, no. 1, pp. 13–29, 2014.
- [18] R. G. Hunt, F. J. Krzystofiak, J. R. Meindl, and A. M. Yousry, "Cognitive style and decision making," *Organ. Behav. Hum. Decis. Process.*, vol. 44, no. 3, pp. 436–453, 1989.
- [19] Lily Shi, H. Bang, G. Hoffman, D. Selva, and S.-Y. Yoon, "Cognitive style and field knowledge in complex design problem solving: A comparative case study of decision support systems," in *Design Computing and Cognition '18*, 2018.
- [20] A. Y. Kolb and D. A. Kolb, "The Kolb Learning Style Inventory — Version 3 . 1 2005 Technical Specifications," 2005.

6. ARCHITECTING DIGITAL TWINS FOR MODEL-CENTRIC ENGINEERING: A COMBINED SEMANTIC MODELING AND MACHINE LEARNING APPROACH – DR. MARK AUSTIN, UNIVERSITY OF MARYLAND

6.1 BACKGROUND

Digital Twins: What’s the problem? How is it done today? Who cares? Challenges? A digital twin is a cyber (or digital) representation of a system that mirrors its implementation in the physical world through real-time monitoring and synchronization of data associated with events. The associated software and algorithms work to provide superior levels of attainable performance in system development and operation. The digital twin concept dates back to the 2000-2010 era, and was initially proposed as a way to support the design and operation of air vehicles for NASA. Since then the range of potential applications has expanded to include automotive components, manufacturing processes, personalized medicine and smart cities, among others.

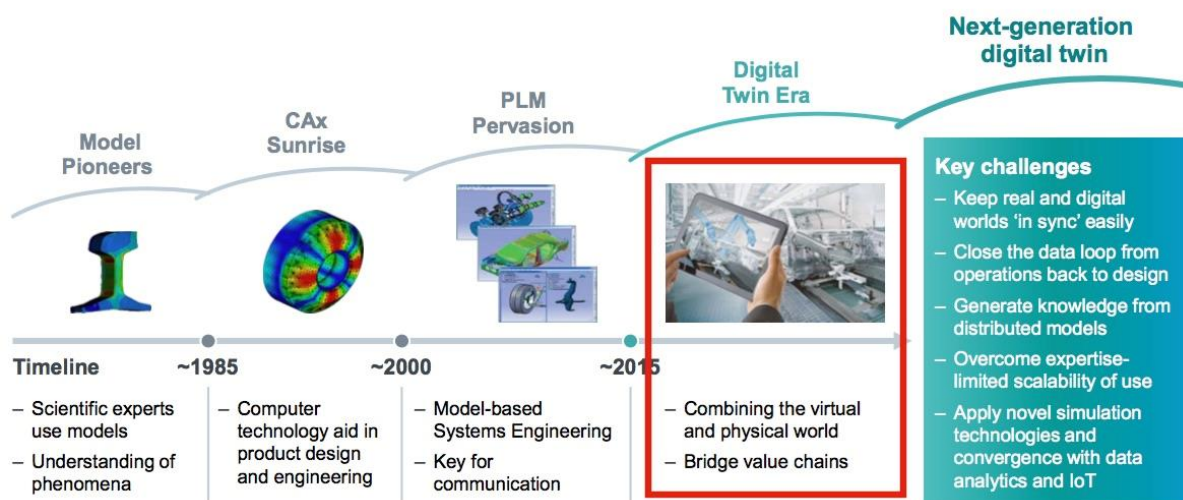


Figure 1. Emergence of Digital Twin Era, a replacement for MBSE with SysML.

Within the world of model-centric engineering, there is strong need for support throughout the entire systems lifecycle, and not just the frontend. As a result (see Figure 1), tool vendors such as Siemens and IBM now anticipate that digital twin capabilities will be the likely successor to model-based systems engineering (e.g., with SysML). Present-day trends in technology development at companies such as Google, Microsoft, Facebook and Apple indicate that AI and machine learning (ML) technologies will be deeply embedded in **new methods and tools** for **model-centric engineering**, as well as new **digital twin operating system environments** for observation, reasoning and physical systems control.

Challenges? Realization of this opportunity is complicated by the reality that within the world of model-centric engineering, present-day use of AI and machine learning technologies is fragmented and at a cross-roads. During that past decade systems engineering researchers in AI (i.e., knowledge representation and reasoning) have tended to focus on the comprehensive development of ontologies for a domain (e.g., satellites) or system development activity (e.g., requirements, system mission, behavior modeling) and their extension from common core ontologies (e.g., for geospatial, time, actors, events) and higher-level basic formal ontologies. Far less attention has been given to the development of rules associated with ontologies, and consideration of the ways in which ontologies and rules can work together to respond to events and support decision making. At the same time, machine learning (i.e., modern neural networks, data mining) techniques provide comprehensive support for the classification, clustering, and identification of association relationships, remembering the details of data streams, and finding anomalies in behavior. Remarkable advances in machine learning algorithms (2016-2019) include the ability of a machine to learn the structure of large-scale graphs and their attributes. Looking forward, the consequences of this recent capability for the model-based systems engineering community would appear to be enormous. And yet, machine learning techniques struggle to explain the rationale for decision making, a task that multi-domain semantic modeling and rule-based reasoning can complete with ease.

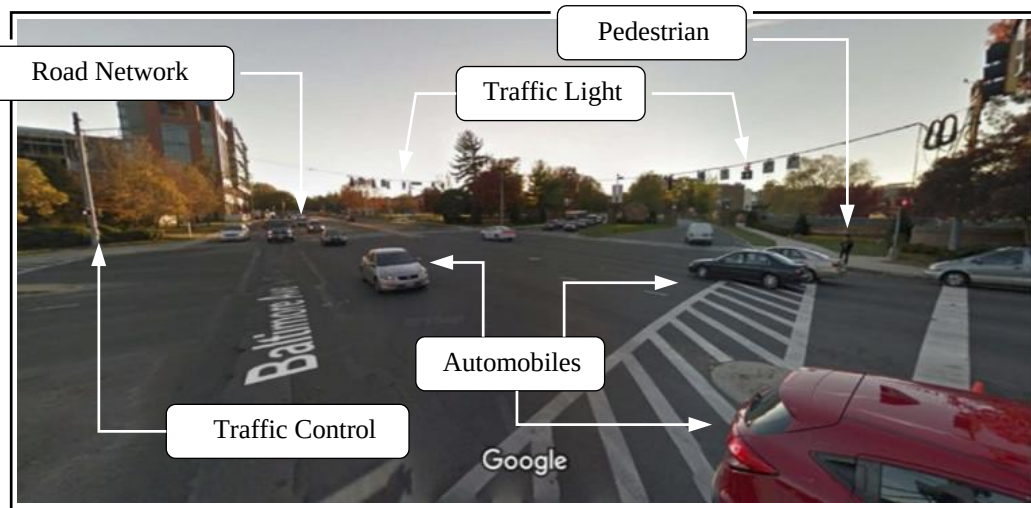


Figure 2. Annotated traffic intersection at the entrance to UMD.

A second source of difficulty stems from enhanced expectations for digital twins enabled by AI/ML technology. Digital twins deployed in real-world situations will be required to produce superior levels of system performance, agility and economy, across multiple scales of problem

size and spatial and temporal extent. Consider, for example, a scenario where a digital twin vehicle has to traverse a busy traffic intersection safely and without causing an accident. As illustrated in Figure 2, challenges include the presence of multiple domains, multiple streams of heterogeneous data, event-driven behaviors, scenarios that are dynamic and time critical. Together with appropriate sensing technologies, AI and ML will be required to observe (monitor) the surrounding environment, evaluate options and take actions in a timely manner. Larger scale systems, such as a city or fleet of aircraft or ships, will be defined by collections of digital twins. From an AI/ML perspective, we expect that individual digital twins will belong to communities, and benefit from AI/ML software common to the community needs.

6.2 PROPOSED WORK FOR POST INCUBATOR RESEARCH

Proposed Work: What's New? Benefits? Will it Work? With these challenges in mind, we propose that the best pathway forward for digital twin design is with architectures that support AI and ML formalisms working side-by-side as a team (see Figure 3), providing complementary and supportive roles in the collection and processing of data, identification of events, and automated decision making throughout the system lifecycle.

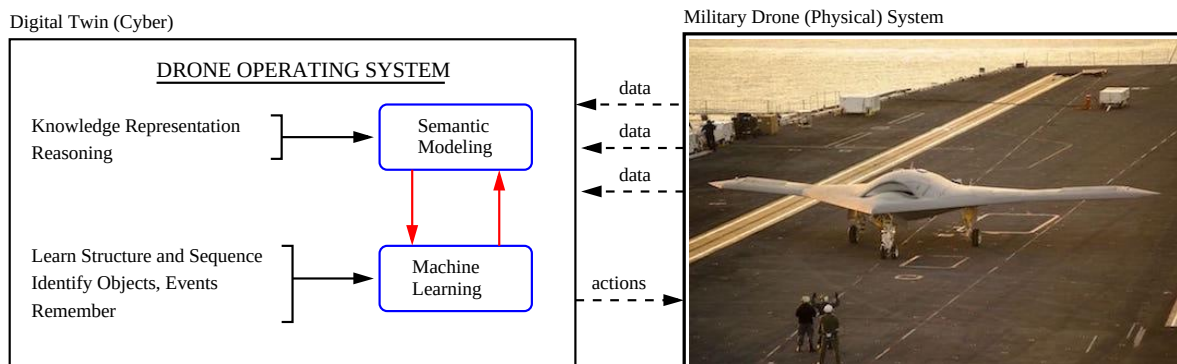


Figure 3. Digital twin (cyber) working alongside a physical system (drone on aircraft carrier).

Research is needed to understand **how to design** the **digital twin elements** and their **interactions** so that collectively they can support two purposes:

1. Development of methods and tools for model centric engineering, and
2. Development of digital twin operating system environments for observation, reasoning and system control.

If successful, this knowledge will be used to guide the architectural development of future digital

twins and threads enabled by AI-ML technology.

Proposed Architectural Template and Preliminary Work (How?). Figure 4 builds upon Figure 3 and shows details of the proposed architectural template components and their interactions.

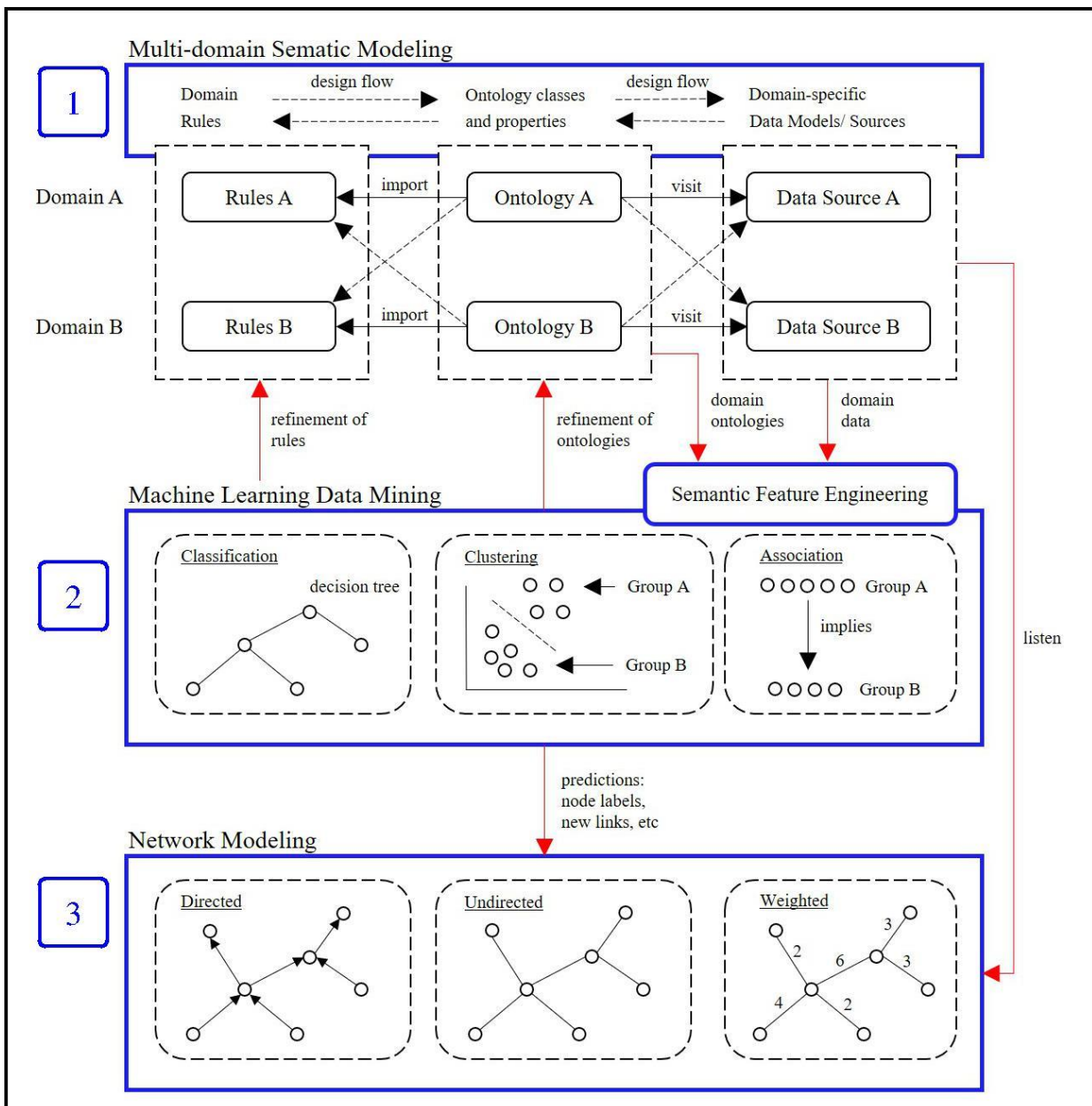


Figure 4. Architectural template for the construction of digital twins. Box 1: Semantic modeling, Box 2: machine learning/data mining, Box 3: Machine learning/network modeling.

This architecture is an extension of our work in 2017 (Delgoshaei and Austin 2017; Coelho, Austin, Blackburn 2018). Box 1 covers a framework for multi-domain semantic modeling and reasoning. The distinguishing feature of this framework is concurrent development of ontologies, rules and data models placed on an equal footing. Box 2 shows machine learning for three classes – classification, clustering and association – of data mining. For the most part, these data mining techniques were developed in the 1980s and 90s. The associated algorithms and software (WEKA, 2019) are now quite mature. During 2018, our studies at UMD focused on semantic foundations and data mining – this is boxes 1 and 2 working together -- for two application areas, energy-efficient buildings and brain cancer profiles (Delgoshaei, Heidarinejad, Austin, 2018; Abraham, Austin, Celiku, 2020), and in both cases, data mining/ML techniques for classification/clustering of data provided useful feedback on the structuring of ontologies (AI side).

Connection to Digital Threads. A closely related, but more difficult, problem is one of using combinations of AI/ML for the modeling of digital threads throughout the systems lifecycle.

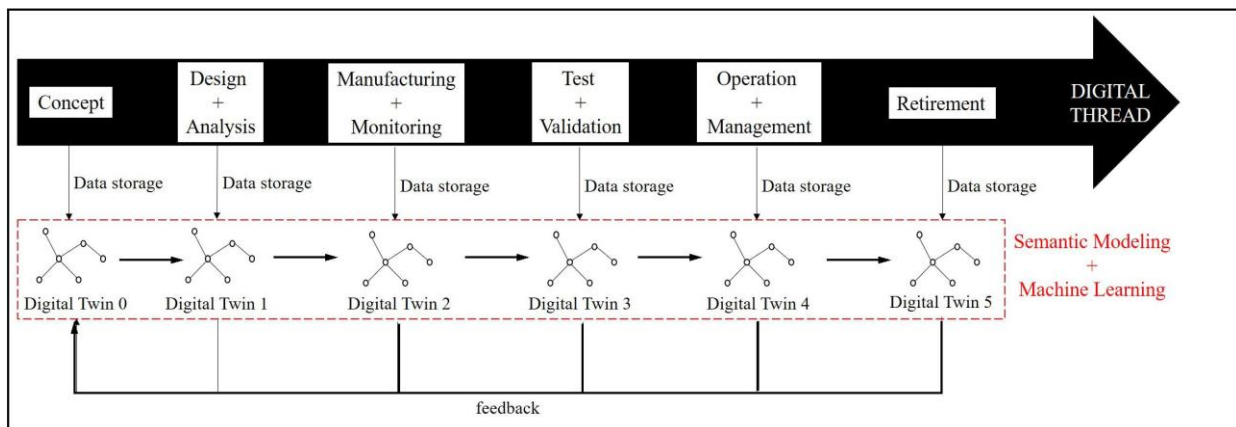


Figure 5. Schematic of a digital thread framework supporting flows of data and integration of viewpoints across the system lifecycle.

A digital thread (see Figure 5) is a communication framework that allows connected data flow and an integrated view of the system’s data throughout its lifecycle, across viewpoints that are isolated functional perspectives. A fully implemented thread enables anticipation and effective communication bi-directionally up and down streams of dependency in the lifecycle. This framework ensures all participants (stakeholders) have access to and can utilize the most current data and can react quickly to changes in the system objectives or in response to new insight. For our purposes, a simple way of distinguishing twins from threads is as follows: twins represent

the state of a system; threads capture the evolution of systems through a sequence of states and transformations. And since systems engineers focus on different things and different stages of the lifecycle – concept, design, manufacturing, test, Operations, retirement – ease of systems adaptability and interoperability is a major challenge in getting digital threads to work.

Getting Started. Focus on AI/ML for Graph Abstractions. While considerable success has already been achieved in understanding semantic modeling + data mining support for methods and tools for model-centric engineering, we still need a good place to start. As illustrated in Figure 5, lifecycle behavior for a digital twin corresponds a sequence of states, from concept to requirement. In fact, a reasonable view of model-centric engineering boils down to representation of systems as graphs and sequences of graph transformations punctuated by decision making and work/actions. In fact, as illustrated in Figure 5, graph abstractions (e.g., activity and Statecharts in SysML; supply chains; models of system structure) are present in all stages of the digital thread lifecycle. From our preliminary work, we already know that multi-domain semantic graphs (AI) can readily adapt to events and rule-driven graph transformations. While this early success shows promise, our current models are far from what seems possible. Thus, the primary purpose of the proposed study is to take the next step (see Box 3): **understand the range of possibilities for which machine learning of large-scale graphs and their attributes can support activities in model-centric engineering.**

Initial Research Questions (Low Hanging Fruit): From a model-centric engineering perspective, there are a whole host of questions that need answering, e.g.:

- What types of graphs (e.g., undirected, directed, weighted) are easy for the ML to learn?
- What can ML techniques do that is outside the capability of semantic modeling? And vice-versa?
- For tasks that can be handled by both the Semantic and ML sides of the framework, what strategies of cooperation make sense?

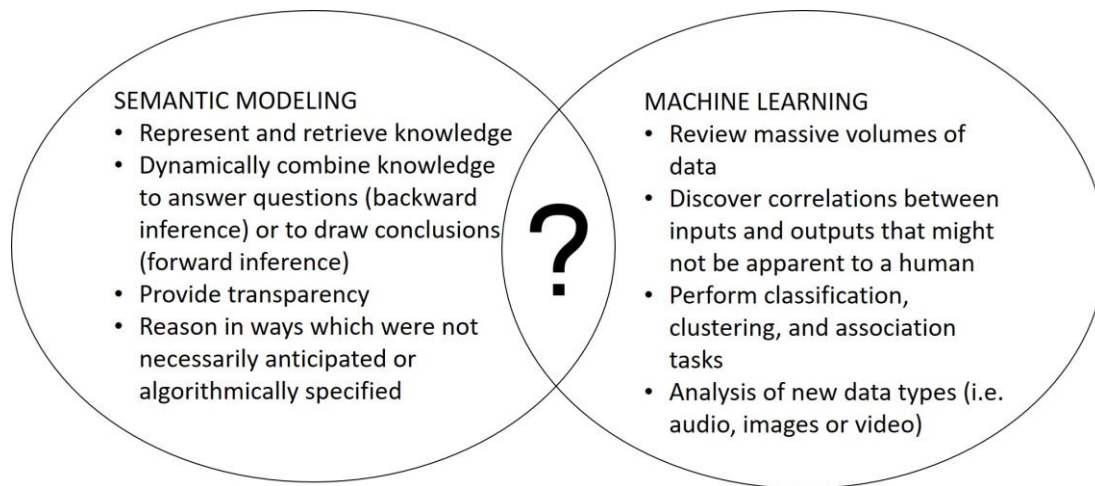


Figure 6. Venn diagram of semantic modeling versus machine learning.

Figure 6 is a preliminary diagram of the key characteristics of semantic modeling versus machine learning, defined by their strengths and weaknesses. It seems to us that the interesting opportunity for cooperation lies with tasks that can be completed by both semantic and machine learning, but for some reason (e.g., attribute values), one side outperforms the other. Such situations hint at new and unique ways AI and ML might work together in win-win scenarios. Looking further ahead (beyond this immediate effort), a well-known limitation of present-day ML techniques is that they can be fragile, with small variations of data causing significant decreases in performance. By having Semantic (AI) techniques work alongside ML, will this combination of formalisms improve system robustness and / or agility?

Companies such as Siemens and IBM are keen to see the digital twin era replace model-based systems engineering with SysML, but in our opinion leaving SysML behind would be a mistake. Thus,

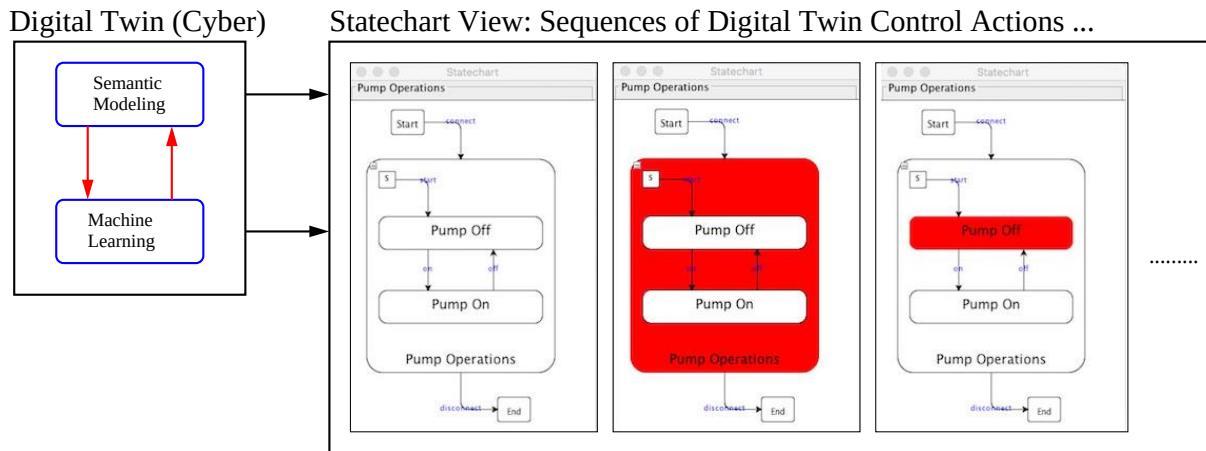


Figure 7. Mapping AI-ML capability to SysML models (not shown) and views.

- How to map AI-ML capability to state-of-the-art engineering views.
- How well do these techniques work with topology and attributes that are dynamic?
- How does the difficulty of these challenges vary as a function of graph size?

It seems to us for very small graphs, the benefits of ML might be very limited? For social networking applications, machine models of graphs having millions of nodes and billions of interactions are now possible.

6.3 ACCOMPLISHMENT OVERVIEW – PROGRESS TOWARDS OBJECTIVES

The results of our Incubator Study can be organized into three parts: (1) Teaching machines to understand graphs (what does that even mean?), (2) Experiments with dynamic attribute network embedding (DANE) vectors, and (3) Opportunities for future AI/ML research.

Part I: Teaching Machines to Understand Graphs. Figure 8 shows simplified representations for traditional and machine learning approaches to graph representation.

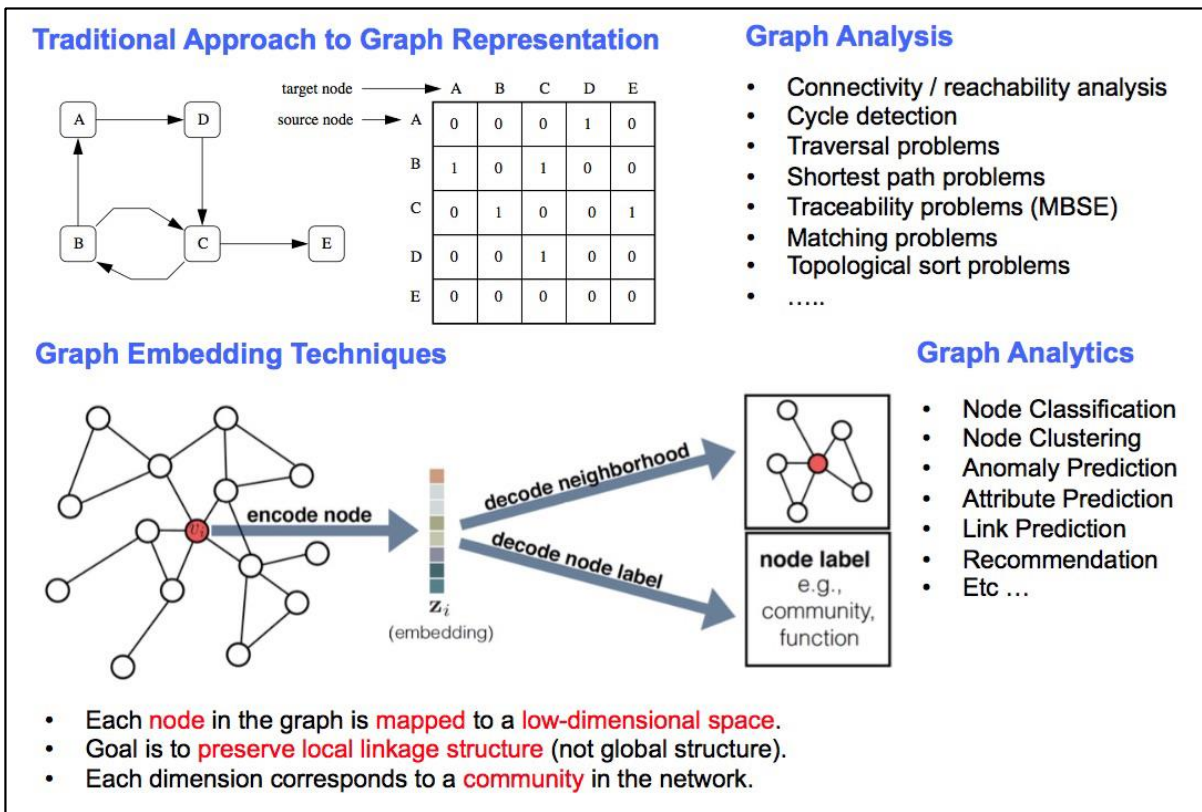


Figure 8. Schematic of traditional (adjacency matrix) and machine learning (graph embedding) approaches to modeling graphs.

Traditional approaches to network/graph modeling employ adjacency matrices (or a simplified representation of network adjacencies) to model the topology of graphs. If one were to build a semantic model of a graph (see Box 1 in Figure 4) the corresponding network ontology would contain classes for the nodes, edges and attributes in a graph and result in a structure along these lines. Mathematical algorithms to determine the properties (e.g., existence of cycles, connectedness, minimum paths) of a graph are well established. This is **graph analysis**.

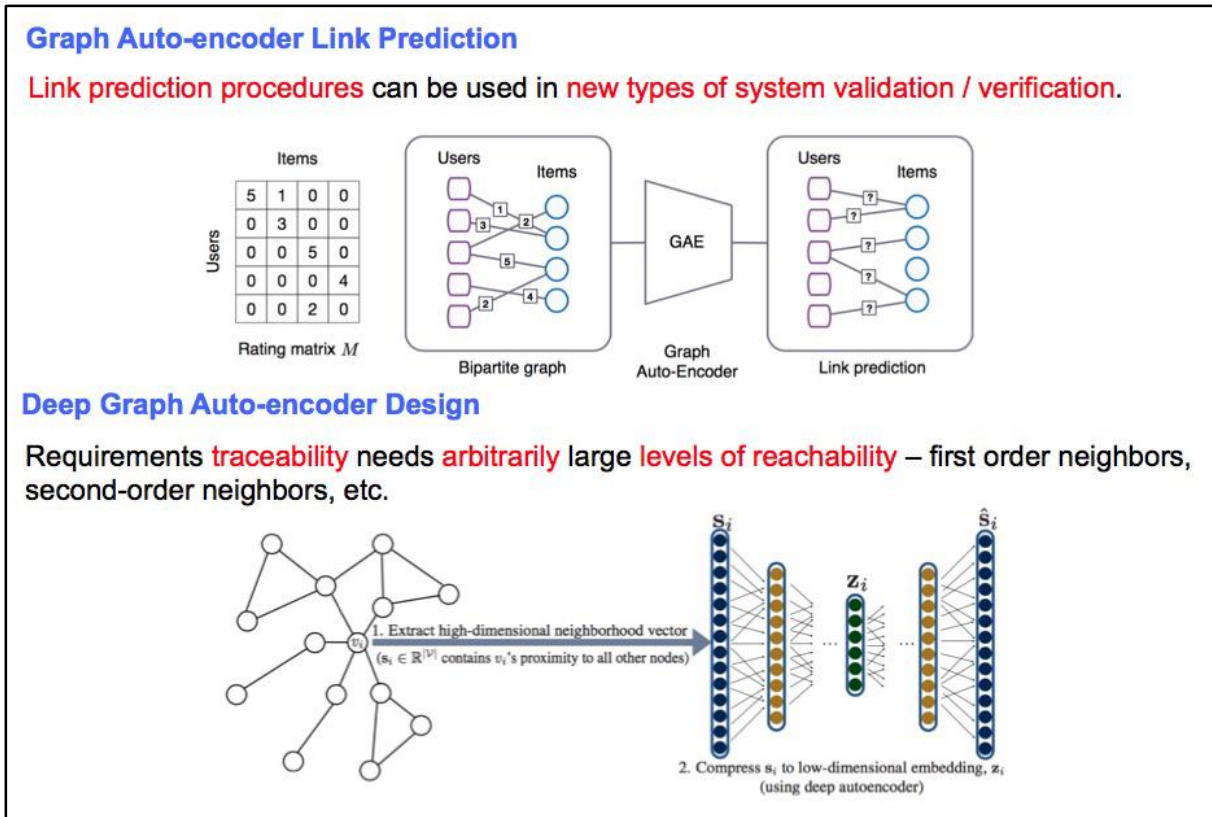


Figure 9. Auto-encoder design for link prediction and deep graphs.

However, for high-dimensional problems that are data sparse, such approaches can quickly become computationally prohibitive. A second problem is that traditional approaches to graph representation do not capture the semantics of the network.

The lower half of Figure 8 shows that simple approaches to machine learning of graphs come at the problem from an entirely different perspective. Instead of focusing on the graph topology (connectivity relations), the graph nodes and their attributes (semantics in domain applications) are mapped (or encoded) to a low-dimensional embedding space, with the goal of **preserving local linkage structure** (not global structure). Embedding structures are derived from random walk-based procedures; three such approaches are LINE, DeepWalk and Node2Vec (Cai, Zheng, Chang, 2018). Decoders are designed to extract views of the graph representation from the low-dimensional embedding. Because information can be lost in the encoder-embedding-decoder transformation process, the output of machine learning for graphs is statistical in nature and, as such, should be interpreted as **graph analytics** (not graph analysis). Graph analytics can support a variety of decision making tasks, including: node classification, node clustering, prediction of anomalies in data streams, link prediction, and recommendations for association relationships.

We believe that with some imagination, these features could be incorporated into new types of tools for model-centric engineering. For example, the upper half of Figure 9 shows an auto-encoder design of link prediction – one can imagine a “missing feature” recommendation service working in parallel with semantics models and rules for system validation and verification. The lower half of Figure 9 shows extensions of the basic auto-encoder design to deep graphs. Notice that unlike the simplified representation of (localized) graph topology shown in Figure 8, the embedding vector for deep graph learning extracts an embedding vector for proximity to all of the other nodes. Thus, one can expect that deep graph representations will contain guarantees on arbitrarily high – first, second, third, forth – levels of nodal proximity. This would appear to be a necessary feature for machine learning to participate in operations for requirements traceability.

Part II. Experiments with Dynamic Attribute Network Embedding (DANE) Vectors. From a model-centric engineering perspective, a key challenge is design of embedding structures that capture the topology of the graph as well as the key characteristics of attribute behaviors associated with graph edges and nodes.

Figure 10 shows how the dynamic attribute network embedding (DANE) architecture works inside a sequence of time steps (Li et al., 2017). Each node is characterized by: (1) its topological relationship to neighbors, and (2) a sequence of feature values. Multi-dimensional tensors of data are assembled for the topological and attribute aspects of the problem formulation, converted into embedding vectors for each concern, and finally merged. The process is repeated for each time step in the simulation behavior.

We have investigated use of the DANE architecture for detecting anomalies in behavior of attribute values in small urban networks (details not included here). A conference paper describing our preliminary results has been submitted to ICONS 2020 and is currently in review (Coelho, Austin, 2020).

Figure 11 is a schematic for how DANE-enabled machine learning might be integrated with simulations for a drone operation. Systems could be trained to understand a number of salient features of acceptable and unacceptable behavior, for example, normal trajectory pathway and cases where a drone drifts off course. Identification of faults could be used as event input the a compatible (AI) semantic model.

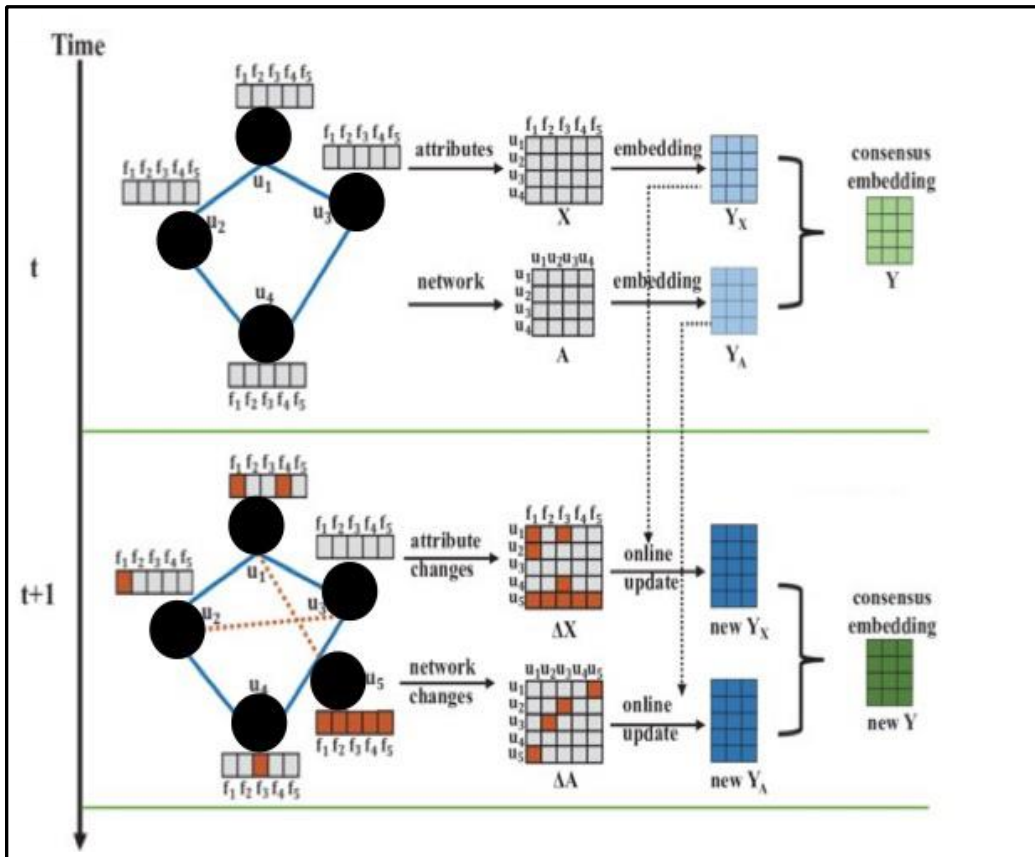


Figure 10. Dynamic attributed network embedding (DANE) Architecture for learning time-history behavior of graph node attributes.

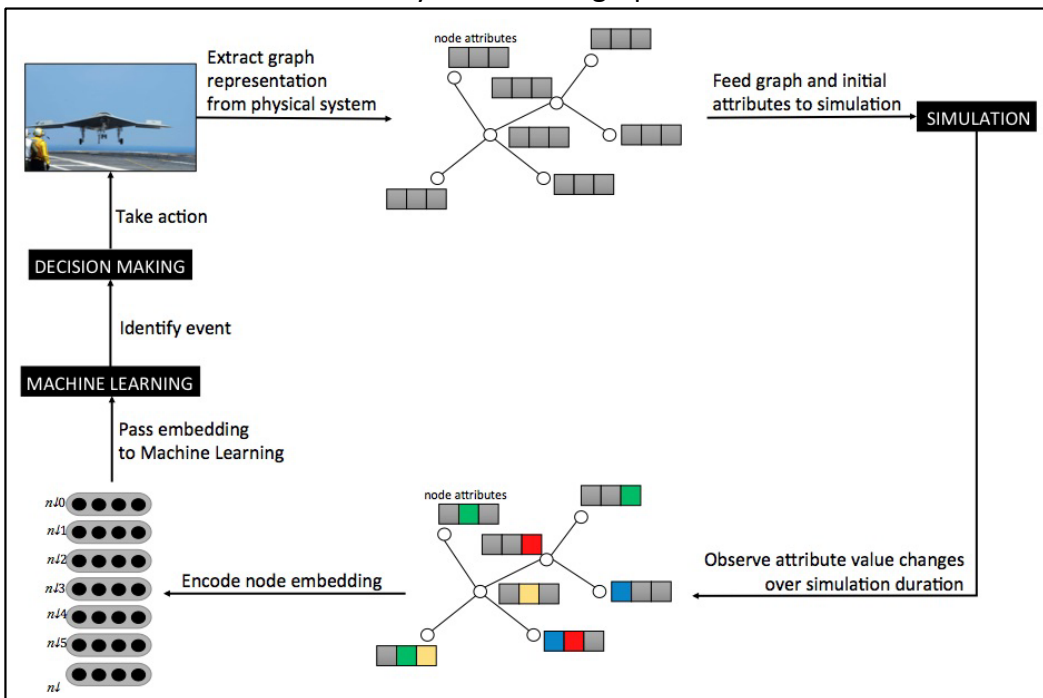
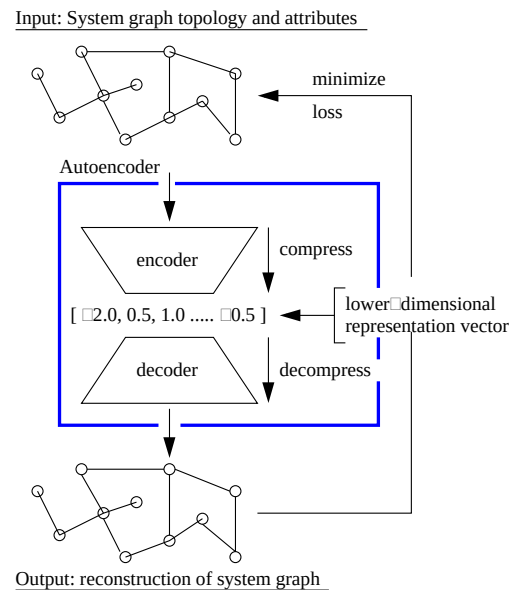


Figure 11. Drone Simulations + DANE-enabled machine learning.

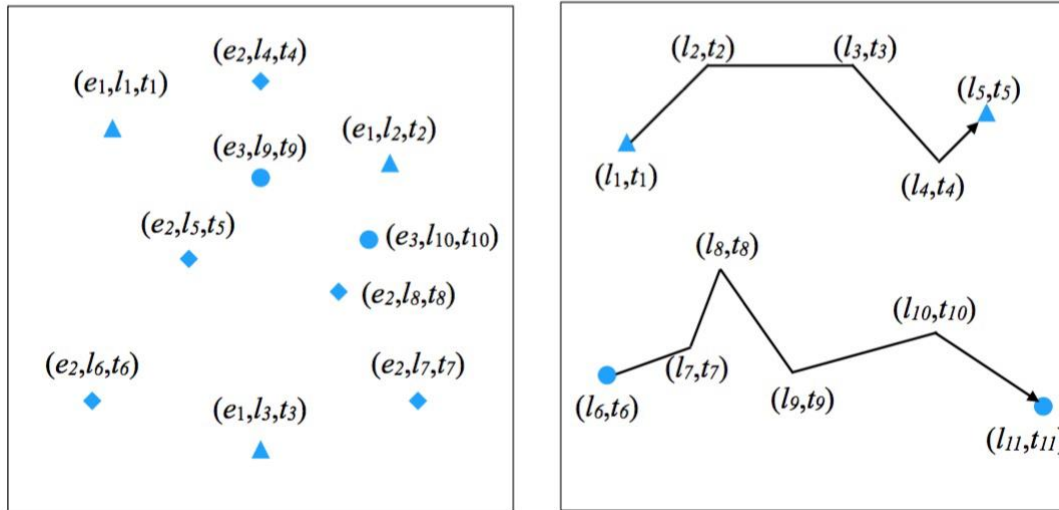
Part III. Extensions and Opportunities: Our preliminary work has exposed a number of opportunities for extensions to the preliminary research:

Topic 1. Precision and Range. Graph auto-encoders (GAEs) are deep neural architectures that map nodes into a feature space and then decode graph information. GAEs can be used to learn network embedding structure (see Figure 8) and decode the topological information for a node. As previously mentioned, first order proximity corresponds to reconstruction of the graph adjacency matrix. Research is need to understand the machine architecture requirements needed to obtain higher – second, third, fourth, and so forth – levels of proximity, such as is needed in applications for traceability of requirements. A closely related issue is one of precision: Are procedures for reconstruction of the graph topology guaranteed to give the correct answer only most of the time or all of the time? During the operational phases of the lifecycle, safety- of life-critical applications will demand the latter.



Topic 2. Injection of Semantics into Auto-Encoder Design. Machine learning algorithms for image classification provide very powerful support for answering questions in a narrow domain, but lack support for placing images in their broader context. For example, convolution neural networks can be trained to identify 100 different types of flow, but they struggle to understand that flowers grow in the ground and only in climates that fall into a certain range. Semantic models, in contrast, are perfect for representing and reasoning with multi-domain knowledge. This observation raises the question: Is it possible to improve the value of machine learning algorithms by injecting semantic information into add-on variables associated with the embedding vectors? Research is needed to understand the extent to which this idea can work.

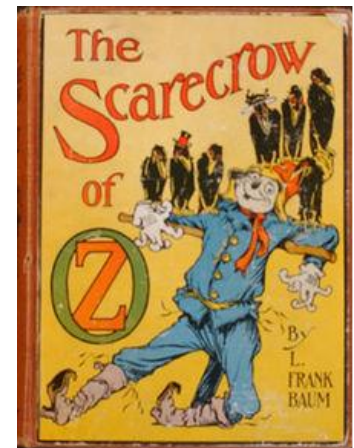
Topic 3. Reasoning with Events, Space and Time. Operational digital twins are mobile and need to make decisions to deal with events in the right place and the right time. Accordingly, we need to develop machine learning capability for reasoning with events, space and time, and linking these capabilities to simulations of military interest.



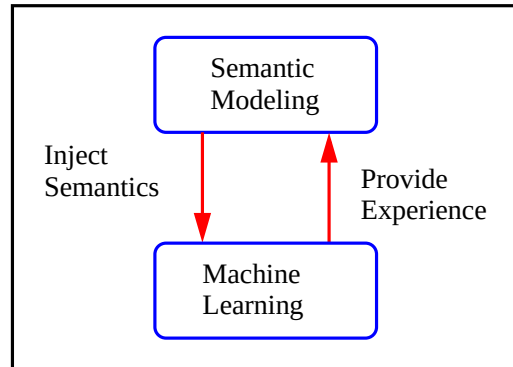
The adjacent figure is adapted from a recent paper by from a research group in Hong Kong (Wang et al, 2019). We need to start following their progress and adapting results to our needs when opportunities arise.

Topic 4. AI/ML Architectures for Digital Twin Experience. In the Wizard of Oz, Scarecrow is a character who cannot remember anything from the past. While he desires above all else to have a brain, in reality he is only two days old and merely naïve. Scarecrow joins Dorothy in the hope that The Wizard will give him a brain. Here, semantic models provide support for multi-domain reasoning that is very much “in the moment” and not based on experience. Every problem is viewed as being new, even if exactly the same problem was solved yesterday. Clearly, this situation is not ideal – surely you want your digital twins to have experience! Other than keeping data property relations around for the duration of the system lifecycle, on the AI side (semantic graph) side of the architecture, there really isn’t a way for semantic models to remember events and decisions from the past. On the other hand, recurrent neural networks have shown remarkable ability to remember the contents of data streams. This leads to the research question: Is it possible to train a machine to record chains of decision making in the semantic model? Such a capability would create a completely new pathway: Design of digital twins with experience, and open doors to improved efficiency in problem solving.

If I only had a brain



Systems Integration of Topics 2 and 4. Together, topics 2 and 4 imply pathways of AI and ML interaction that are mutually supportive and beneficial. By injecting semantics into layers of the ML encoder, or perhaps the embedding vector, an opportunity exists to broaden the scope of scope of understanding ML processor will have of a domain. Conversely, by exploring ways that ML might assist the semantic modeling through recording of past scenarios and pathways of decision making, efficiency of decision making on the semantic side will be enhanced. This mutually beneficial relationship is illustrated in the adjacent figure.



6.4 AGENDA AND BUDGET FOR PROPOSED RESEARCH

Our agenda for proposed research is as follows:

Year 1. Teaching Machine to Understand Graphs

- Teaching machines to understand small graphs having static graph topologies.
- Auto-encoder design (guarantees on system graph representation).
- Model directed graphs with deep convolution neural networks.
- Identification of events via time-series anomaly detection.
- Basic mechanisms for semantic / machine learning interaction.
- Integration of simulation and machine learning.

Year 2. Go Deep, Dynamic, Broad and Hybrid

- Deep graph neural networks / dynamic graph topologies.
- Reasoning with events, space and time.
- Map AI-ML to state-of-the-art views (e.g., SysML).
- Injection of semantics into machine learning models.
- Applications.

Year 3. Create Digital Twin Experience

- AI/ML architectures for digital twin experience.
- Applications.

UMD Budget: \$250 k/yr.

6.5 REFERENCES

1. Coelho M., Austin M.A., and Blackburn M.R., The Data-Ontology-Rule Footing: A Building Block for Knowledge-Based Development and Event-Driven Execution of Multi-Domain Systems, 2018 CSER, Charlottesville, VA, May 8-9, 2018.
2. Delgoshaei P. and Austin M.A. , Framework for Knowledge-Based Fault Detection and Diagnostics in Multi-Domain Systems: Application to Heating Ventilation and Air Conditioning System, International Journal on Advances in Intelligent Systems, Vol. 10, No 3 and 4, December 2017, pp. 393-409.
3. Delgoshaei P., Heidarinejad M., and Austin M.A., Combined Ontology-Driven and Machine Learning Approach to Management of Building Energy Consumption," 2018 Building Performance Analysis Conference and SimBuild, Chicago, September 26-28, 2018.
4. Abraham J., Austin M.A., and Celiku O., Mining High-Dimensional Glioma Data for Ontologies and Rules, The Fifteenth International Conference on Systems (ICONS 2020), Lisbon, Portugal, February 2020 (Submitted).
5. Coelho M., and Austin M.A., Teaching Machines to Understand Urban Networks, The Fifteenth International Conference on Systems (ICONS 2020), Lisbon, Portugal, February 2020 (Submitted).
6. WEKA (Waikato Environment for Knowledge Acquisition). Open source Java software for data mining. See: <https://www.cs.waikato.ac.nz/weka/>
7. Wang S., Cao J., Deep Learning Models for Spatio-Temporal Data Mining: A Survey, arXiv, June 24, 2019.
8. Li J., Dani H., Hu X., Tang J., Chang Y. and Liu H., Attributed Network Embedding for Learning in a Dynamic Environment, In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, ser. CIKM '17. New York, NY, USA: ACM, 2017, pp. 387–396. [Online].
9. Cai H., Zheng V.W., and Chang K.C., "A Comprehensive Survey of Graph Embedding: Problems, Techniques and Applications," IEEE Transactions on Knowledge and Data Processing, vol. 30, no. 9, 2018, pp. 1616–1637. ^[1]_{SEP}

7. ECOLOGY-INSPIRED TECHNIQUE FOR RESILIENT ENGINEERED SYSTEM OF SYSTEMS DESIGN – DR. RICHARD MALAK, TEXAS A&M UNIVERSITY

ABSTRACT

The research team investigated a bio-inspired mathematical framework for analyzing and predicting trade-offs between System of Systems (SoS) performance, affordability, and resilience early in the design process without the need for highly detailed simulations or disruption models. This framework builds on a body of ecological research that has found a unique balance between redundancy and efficiency in biological ecosystems. This balance implies that highly efficient ecosystems tend to be inflexible and vulnerable to perturbations whereas highly redundant ecosystems fail to utilize resources effectively for survival. A pilot study was conducted to demonstrate the effectiveness of the mathematical framework. Twenty architectures for a notional hostiles surveillance SoS were investigated. Results indicate that highly efficient SoS architectures fail catastrophically in the face of disruptions while highly redundant architectures are unnecessarily expensive. These results support the idea that engineered SoS architectures follow a *fitness* trend akin to complex ecological networks and suggest that SoS engineering can benefit from considering a balance of redundancy and efficiency similar to that found in ecological networks.

7.1 INTRODUCTION

Successfully achieving mission objectives requires the intelligent utilization of all participating systems. The term ‘System of Systems’ (SoS) is used to describe a collection of independently operating systems, which interact with one another to accomplish mission requirements that cannot be achieved by the individual systems alone [1, 2]. These systems may provide essential capabilities such as surveillance, detection, information processing and exploitation, neutralization of hostiles, and the supply of aid, all within the larger SoS context. SoS have evolutionary tendencies (addition of new systems over time), a range of geographic distributions, and operational independence of participating systems [3]. Additionally, a functional interdependence of the systems within the SoS creates challenges for their analysis and development.

The attributes **performance**, **affordability**, and **resilience** [4-6] are most commonly used to analyze and track SoS development. Performance, as used here, quantifies the extent to which an SoS meets mission objectives. Affordability is taken here to account for the capital and operational costs of an SoS. Resilience is defined as the ability of an SoS to survive and recover from disruptions [5, 7] and is of particular importance when operations are in high risk

environments. The best SoS architectures are the ones that balance the trade-offs between these attributes, based on the mission objective and the operating environment.

7.1.1 Research Objective

The goal of this exploratory research is to investigate a bio-inspired framework for analyzing and predicting trade-offs between System of Systems (SoS) performance, affordability, and resilience early in the design process – without the need for highly detailed simulations or disruption models.

7.1.2 Limitations of Current Practice



Figure 1: Simplified design for resilience process, based on [5].

Design for resilience is an iterative process: Engineers make design decisions to improve a system’s (or SoS) resilience to selected disruptive events and then evaluate the response of the system (or SoS) to the disruption. Based on the resilience evaluation, engineers revisit design decisions, make the required changes, and the cycle continues till a favorable/acceptable design is found. There is no single measure that captures engineered resilience. Instead, engineers capture aspects of resilience using various metrics (survivability, recovery time, recoverability, etc.) [8, 9]. Most of these entail an analysis or simulation conducted using a detailed description of the SoS in question and specific knowledge of potential disruptions to the SoS.

Currently, only general design principles exist to guide engineers toward resilient SoS solutions, like physical redundancy, functional redundancy, localized capacity (amongst others) [10]. However, these are only qualitative guides and do not provide insight into how much redundancy or distribution of capacity is “enough” and how much is “too much?” Due to the wide geographic distribution of the participating systems in an SoS, increasing redundancy can be costly [5]. Excessively redundant architectures can also result in interoperability or interference issues (especially in SoSs with diverse participating systems) [11, 12], resulting in loss of performance.

7.1.3 Inspiration – Resilient Ecosystems

Ecosystems are biological SoS that are resilient to a wide variety of perturbations [13]. They are made up of independent actors (species or functional groups) that perform essential roles in the network (like producers, predators, and decomposers) and have an evolutionary nature. This

investigation builds on a body of ecological research that has found a unique balance between redundancy and efficiency in biological ecosystems. This balance implies that highly efficient ecosystems tend to be inflexible and vulnerable to perturbations, while highly redundant ecosystems fail to utilize resources effectively for survival. The Ecological Network Analysis (ENA) metric *degree of system order* (a) indicates the tradeoff between the level of organization (constraints/pathway efficiency) in a flow network and the organizational capacity of the network (flexibility/redundant pathways). Long-surviving complex ecosystems have been found to exist within a specific range of a – dubbed the *window of vitality*.

This research models and analyzes a notional hostiles surveillance SoS architectures as a network of interacting systems to gather evidence about whether the ecologically inspired concept of the “window of vitality” applies to engineered SoS. Do SoS architectures with bio-inspired balances of efficient and redundant interactions present better performance, affordability, and response to disruptions? Factors that govern an engineering SoS “window of vitality” are expected to be clarified when compared to highly efficient or redundant architectures.

7.1.4 Value Proposition

The analysis of the degree of system order only requires knowledge of network architecture: the participating actors and their interactions (the mathematical framework is discussed in section 2). If evidence is found that, akin ecosystems, engineered SoSs with optimal attributes trade-offs also exist in specific ranges of the degree of system order metric, it can transform how engineers approach SoS development. As a tool for system engineers, the degree of system order metric could enable selection between SoS architectures for those with the best attributes (such as performance, affordability, and resilience) without the need for highly detailed simulations or disruption models.

7.2 ENA, THE DEGREE OF SYSTEM ORDER, AND THE ECOLOGICAL FITNESS OF FUNCTION

Ecologists use ecological network analysis (ENA) to study the complex interactions among the species within a food web (an ecosystem modeled in terms of its predator-prey based interactions). A directional graph or digraph is created for the food web, where the nodes represent the actors or species, and the directed arcs represent the transfer of energy or nutrients between them and their immediate environment. Flow magnitude information between the nodes within the system boundaries, as well as those with the surrounding environment (system inputs, outputs, and dissipations), are all stored in the square $(N+3) \times (N+3)$ flow matrix $\mathbf{T}$ (where N is the number of actors within the system or SoS boundaries). The matrix elements T_{ij} represent the magnitude of flow from node i (producers/prey) to node j (consumers/predators). The nodes 1 to N in the flow matrix represent the actors within the

specified system boundary. The nodes 0, $N+1$, and $N+2$ are the system imports (row “zero” in the **T** matrix) and system exports & dissipations (columns $N+1$ and $N+2$, respectively). Fig. 2 illustrates this process, with a notional food web (top-left) modeled as a digraph (bottom-left) and then quantified in its flow matrix **T** (bottom-center). Readers interested in a more detailed description may refer to *Fath et al. (2007)* [14].

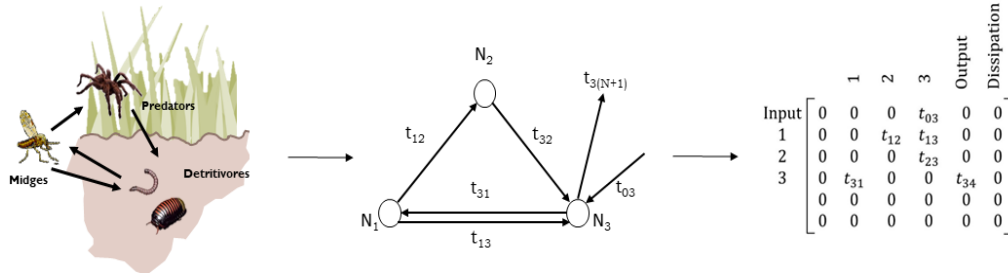


Figure 2: A schematic of the modeling procedure used in ENA, based on *Layton et al. (2015)* [15].

ENA metrics are calculated using the **T** matrix. The sum of all the flows through the network is the *Total System Throughput* ($TSTp$, Eq. 1). *Ascendancy* (ASC) measures the network’s organizational development, or the ability of the network to efficiently transport the medium of interest from one point to another. When normalized by $TSTp$, ASC becomes *Average Mutual Information* (AMI , Eq. 2) [16]. Nothing can grow in nature without bounds, including network organization. The upper limit on ASC is the *Development Capacity* (DC). *Shannon Index* (H , Eq. 3) is DC normalized by $TSTp$. These metrics are rigorously derived using the concepts of information theory in scholarly works by Ulanowicz [16, 17].

$$TSTp = \sum_{i=0}^{N+2} \sum_{j=0}^{N+2} T_{ij} \quad (1)$$

$$AMI = \sum_{i=0}^{N+2} \sum_{j=0}^{N+2} \frac{T_{ij}}{TSTp} \log_2 \left[\frac{T_{ij} \cdot TSTp}{T_i \cdot T_j} \right] \quad (2)$$

$$H = - \sum_{i=0}^{N+2} \sum_{j=0}^{N+2} \frac{T_{ij}}{TSTp} \log_2 \left[\frac{T_{ij}}{TSTp} \right] \quad (3)$$

Where,

$$T_i = \sum_{j=0}^{N+2} T_{ij} \quad (4)$$

$$T_j = \sum_{i=0}^{N+2} T_{ij} \quad (5)$$

These metrics are used for networks with only one medium or unit of flow. Engineered SoSs consist of *multiple* distinct and interdependent flows, however, and the translation of these metrics requires their modification to allow for multiple flows: AMI and H in Eq. 6 and 7 have been reformulated to accommodate multiple flows [18]. The symbols retain their original meaning, and the additional subscript l signifies the different types (mediums) of flows where $l \in [1, M]$ for a network composed of M distinct flow types.

$$AMI = \sum_{i=0}^{N+2} \sum_{j=0}^{N+2} \left\{ \left(\prod_{l=1}^M \frac{T_{ijl}}{TST_{p_l}} \right) \log_2 \left[\frac{\prod_{l=1}^M T_{ijl} \cdot TST_{p_l}}{\prod_{l=1}^M T_{i,l'} \cdot T_{j,l'}} \right] \right\} \quad (6)$$

$$H = - \sum_{i=0}^{N+2} \sum_{j=0}^{N+2} \left\{ \left(\prod_{l=1}^M \frac{T_{ijl}}{TST_{p_l}} \right) \log_2 \left[\prod_{l=1}^M \frac{T_{ijl}}{TST_{p_l}} \right] \right\} \quad (7)$$

The ratio of AMI to H is the *degree of system order* (a) [17] and has values ranging from zero to one. A value of a close to zero indicates that a large number of redundant pathways exist in the network, and a value close to one indicates that the network has a minimum number of highly constrained (or efficient) flow pathways. These extreme cases correspond to networks where either all actors are connected to each other (most redundant, $a = 0$) or all actors are connected in a linear series (highest efficiency, $a = 1$). Neither of these extreme cases results in a ‘fit’ network. An excessively redundant network is not effective at utilizing the available resources, and an excessively efficient network will be vulnerable to disruptions.

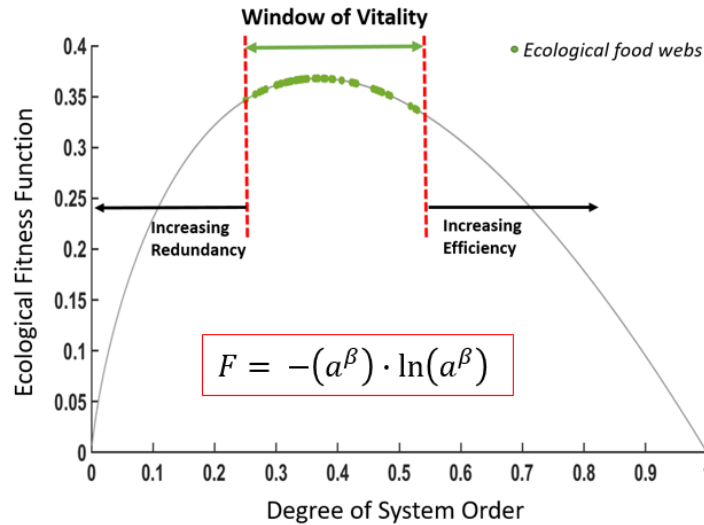


Figure 3: The ecological fitness (R_{ECO} , Eq. 9) curve with food webs (green dots) from the dataset of Borrett et al. [19] illustrating the ‘window of vitality’ [17].

Fit, or optimal, network architectures can safely be assumed to lie between these two extremes. This is mathematically reflected in Ulanowicz's formulation of the *fitness function* of a flow network (F), which is a function of the *degree of system order* and the *Boltzmann measure of its disorder* ($-k \cdot \ln(a)$) [17, 20]. Equation 8 is the general form of this fitness function. The range of a where mature ecosystems reside, known as the "window of vitality," indicates that complex systems in nature have evolved to a selection of $a \sim 1/e$ [17]. Equation 9 is the resulting *ecological fitness function*, plotted in Fig. 3 with 48 different ecosystems residing near $a \sim 1/e$.

$$F = -(a^\beta) \cdot \ln(a^\beta) \quad (8)$$

$$R_{eco} = -(a) \cdot \ln(a) \quad (9)$$

7.3 INVESTIGATING THE *FITNESS* TRENDS IN A NOTIONAL HOSTILES SURVEILLANCE SoS

Twenty feasible architectures of a hypothetical hostiles surveillance SoS are investigated here, consisting of a mission command center, the Continental United States headquarters (CONUS), and on-site surveillance and data exploitation systems. The objective of this SoS is to monitor a 9000 sq. miles area continuously for any sign of hostile activity and then select and implement appropriate response measures. The available on-site surveillance systems are Joint Surveillance and Target Attack Radar (JSTAR) aircrafts, Unmanned Aerial Vehicles (UAV), and a military satellite. Every two UAVs require a local, on-ground control and exploitation unit (theater) while JSTAR has onboard crewmembers to perform data exploitation [21]. CONUS handles the control and data exploitation of the satellite and provides the mission command with the exploited data. CONUS also exchanges commands and reports with the mission command center and acts as the mediator between government agencies and the SoS. The mission command center receives exploited surveillance data from all participating systems, monitors their status, and commands their operations.

Table 1 shows the performance characteristics and operational costs of the systems within this hypothetical SoS based on descriptions found in [6]. The satellite is assumed to already be in orbit and operational, only needing to be commissioned for the mission, resulting in a low operational cost [6]. All surveillance is assigned a quality value between zero and one for the precision and clarity of the raw data collected.

Table 1: Performance characteristics for the available systems within the hostiles' surveillance SoS based on descriptions found in [6].

System	Surveillance quality	Max. surveillance area (sq. miles)	Exploitation capability	Operational Cost (10^3 \$/hour-unit)
JSTAR	1	>9000	1 JSTAR	18
UAV	0.9	2250	-	2
Military Satellite	0.7	>9000	-	1
Theater	-	-	2 UAVs	10
CONUS	-	-	Unlimited	-

7.3.1 SoS performance, cost, and response to disruptions

The total operational cost of the SoS is calculated as the sum of the operational costs of each participating system. The performance level of the SoS is formulated as shown in Eq. 10 and is based on [7]. A is the area covered by a surveillance system (in sq. miles), q indicates the quality of surveillance by that system, s indicates the state of the exploitation system associated with the surveillance unit, and the subscript i indicates the systems under consideration. The maximum surveillance performance level is 9000, for coverage of the entire area of interest.

$$Performance = \max[(\sum_1^N A_i \cdot q_i \cdot s_i), 9000] \quad (10)$$

The loss of one, two, and three systems in the SoS (N-1, N-2, and N-3 contingencies) were randomly simulated to investigate the response of the SoS architectures to external disruptions. The performance level of the SoS right after the loss of its system(s) as well as after the mission command re-allocates the surviving systems was then calculated. These contingency analyses consider any on-site units as candidates for attack by hostiles. The satellite is assumed to have partial immunity to hostiles, resulting in a loss of only communications. The mission command center and CONUS are assumed to be safe from external disruptions.

7.3.2 Degree of system order of SoS architectures

The participating systems have two kinds of interactions: (a) the flow of surveillance data and (b) the exchange of commands and reports. The magnitude of surveillance data collected by surveillance units is assumed to be the product of the area they cover and the surveillance quality of the system. 10% of the raw surveillance data collected is assumed useful after exploitation. Fifty (50) units of command and reports data are assumed to be exchanged between the mission command and each surveillance/exploitation system. Three hundred (300) units of commands and reports data are assumed to be exchanged between the mission command, CONUS, and governing authorities.

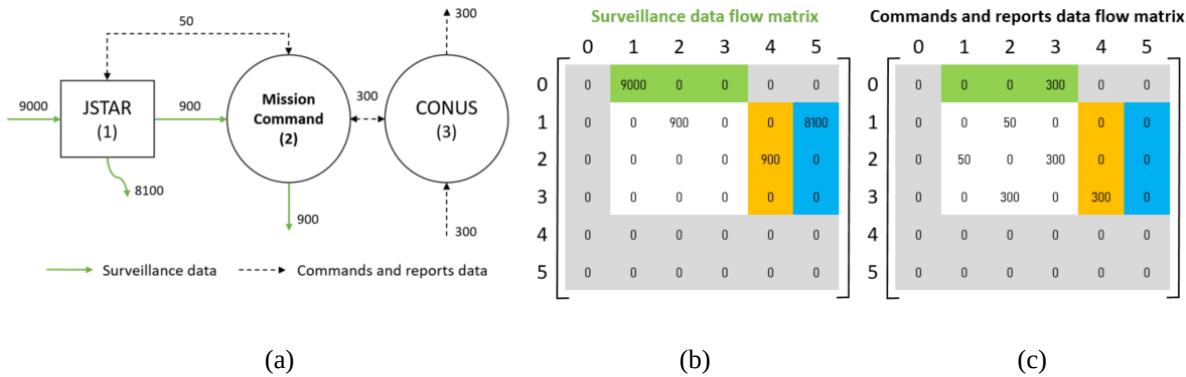


Figure 4: A schematic of the surveillance SoS represented as a data flow network (the numbers on the directed arcs represent the magnitude of data flows) and the associated (b) surveillance data flow matrix and (b) commands and reports data flow matrix.

Figure 4 shows the case study SoS architecture modeled as a flow network and the associated flow matrices. The JSTAR imports surveillance data through imaging (entry T_{01} in Fig. 4b), exploits the raw data, and provides useful data to the mission command center (entry T_{12} in Fig. 4b). The remaining (non-useful) raw surveillance data is modeled as dissipation (entry T_{15} in Fig. 4b). Mission command utilizes the exploited data for decision-making, which is modeled as a useful export (entry T_{24} in Fig. 4b). There is also a bi-directional flow of commands and reports between mission command and the JSTAR (entries T_{12} and T_{21} in Fig. 4c), as well as CONUS and mission command (entries T_{32} and T_{23} in Fig. 4c). CONUS receives orders from the governing authorities outside the SoS boundary (import, entry T_{03} in Fig. 4c) and reports back to them (useful export, entry T_{34} in Fig. 4c). AMI and H were calculated using Eqs. 6 and 7 with the information from the two matrices in Fig. 4c and 4d. The degree of system order was then calculated using the AMI and H values.

7.4 RESULTS AND DISCUSSION

The twenty SoS architectures investigated utilize the techniques of physical redundancy (e.g., multiple JSTARs), functional redundancy (e.g., surveillance using both JSTAR and the satellite), and localized capacity (e.g., sharing the surveillance load between multiple drones). Degree of system order values ranged between 0.3 to 0.65.

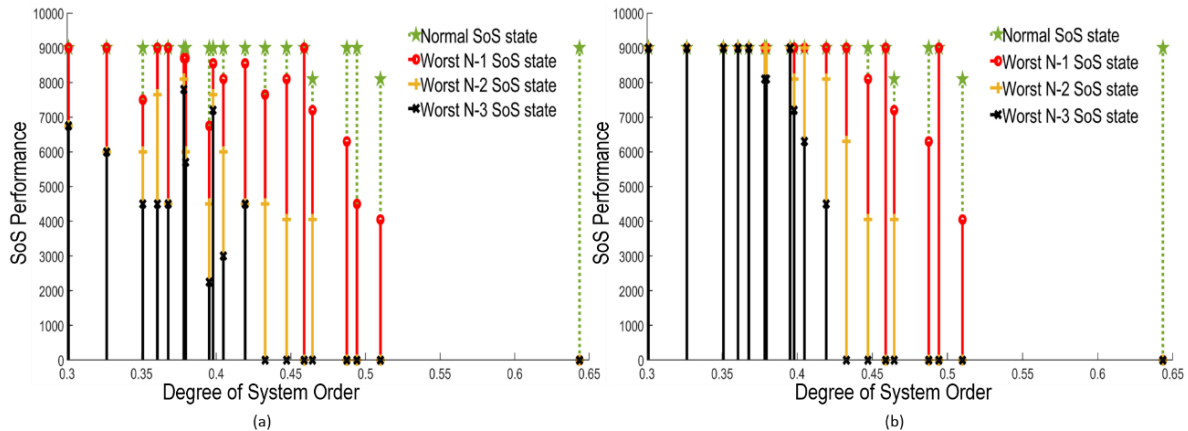


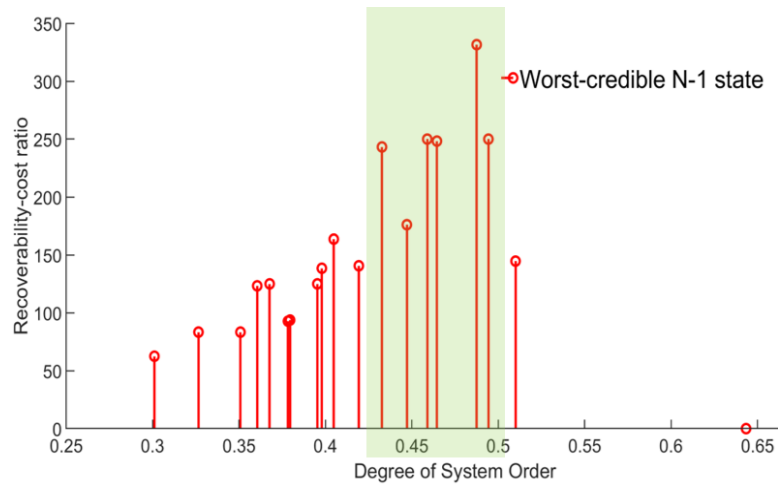
Figure 5: The worst-credible SoS performance levels (a) immediately after the disruption and (b) after re-allocation of surviving systems.

Fig. 5. shows the SoS performance levels immediately after the worst possible $N-1$, $N-2$, and $N-3$ disruptions (Fig. 5a) and after re-allocation of surviving systems (worst-credible states, Fig. 5b). Networks highest in pathway efficiency suffered complete failures and could not regain any functionality. Their lack of redundancy created an inability to recover the lost functionality. The SoS architecture with redundancy can be seen to have the flexibility needed to survive and recover from external disruptions. A tipping point, where additional redundancy no longer improves the SoS resilience, is seen though in Fig. 5. This point is closer to one (efficient architecture) for the $N-1$ scenarios and shifts towards $a = 0$ (redundant pathways) for the more severe $N-2$ and $N-3$ scenarios.

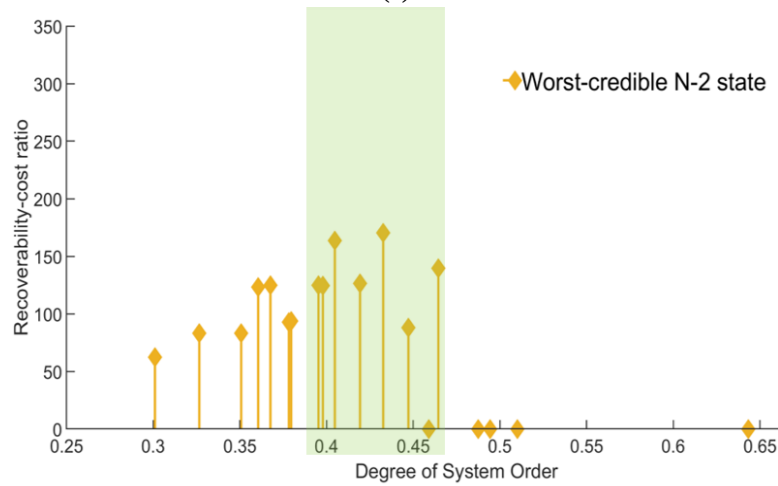
The behavior of ecosystems offers a potential route to answer the question - *how much redundancy is enough vs. too much?* The case-study here is inadequate to analyze interoperability issues but can analyze the ratio of recovered performance to operational cost against the degree of system order (Fig. 6). It is observed (see Fig. 6) that the *recoverability to cost ratio* of the notional SoS architectures first improved with more redundancy (lower a values) until it reached a peak, and then decreased with additional redundant interactions. This overall behavior imitates the fitness of complex systems observed in ecology, evidence that SoS architectures can learn to

balance efficient and redundant interactions from ecology to better manage their performance, affordability, and response to disruptions.

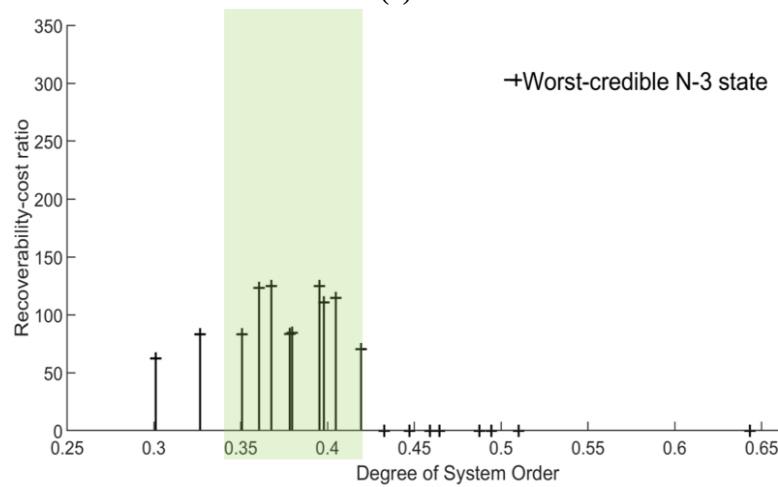
The results also showed that each N - X case has a select number of designs covering a slightly different α range that have significantly better recoverability-cost ratios (indicating a better trade-off between affordability and response to disruptions). The range of the favorable α values seems stricter than the ecosystems' *window of vitality*. This may be due to the ecosystem data covering biological SoS that experience a range of disturbances, from mild to extreme, while the notional SoS investigated here has been grouped by specific N - X disturbance levels. The *fittest* SoS architectures can be seen to favor values of α closer to 0 as the severity of disruptions increases (with increasing X values in N - X). This observation indicates that higher threat levels cause the SoS fitness trends to move towards the *ecological fitness function*, suggesting that the level of threats in an SoS's operating environments may provide a governing factor in deciding its specific *window of vitality*.



(a)



(b)



(c)

Figure 6: The recoverability-cost ratios in the worst-credible (a) N-1, (b) N-2, and (c) N-3 disruptions of the twenty SoS architectures plotted against their degree of system order. The *window* of the *fittest* SoS architectures is

highlighted in each case. With increasing severity of disruptions, the *window* of the *fittest* SoS architectures shifts towards more pathway redundant architectures

7.5 CONCLUSIONS

Evidence from this SERC Incubator project supports the idea that the ecology-inspired concept of the *window of vitality* can be useful for the design and management of resilient engineered SoS. This can fill a significant gap in current practice. Known design rules provide useful heuristics for architecting resilient SoS, but these constitute only rough guidelines for engineers. SoS simulations allow quantitative analyses of resilience and other properties, but require detailed models of the system and specific threats/disruptions. The ecology-inspired mathematical framework investigated here can enable quantitative analyses earlier in the SoS engineering process.

Although promising, further study of the modeling framework is needed to create a useful SoS engineering tool. For example, we found the range of α values presenting favorable SoS attribute trade-offs is stricter than for natural ecosystems, which presents an opportunity to customize the window of vitality for an SoS based on its specific needs and environment. This could lead to SoS design guidance with different qualitative threat level categories (e.g., low, medium and high) corresponding to different vitality windows. More complex SoS case studies would provide evidence needed to validate these ideas and create a useful engineering tool for selecting SoS architectures without the need for costly simulations or detailed disruption models.

7.6 FUTURE WORK

7.6.1 Extensions to Mathematical Framework

The SERC Incubator project focuses on a specific construct from the ecological network analysis community, called the *window of vitality*. Results indicate this is relevant for engineering resilient SoS, but the full potential of the mathematical framework associated with ecological network analysis has yet to be explored. Open issues include the following:

- Underlying Relationships. Incubator results suggest that fundamental relationships may exist that help explain why a window of vitality should exist and how to think about this construct in a practical way during an engineering project. For example, we noted a trend between disruption severity (in terms of SoS elements disabled) and the window of vitality location. A more complete mathematical framework would capture this underlying relationship explicitly (i.e., explain the peak fitness, β , parameter and range of favorable α values). We also speculate that affordability is not the only impediment to further increases in SoS

redundancy. It would be worthwhile to investigate the effect of excessive redundant interactions in the SoS architecture on SoS resilience.

- Useful Metrics and Design Guidelines. The Incubator project established the relevance of the window of vitality construct, but other metrics associated with ecological network analysis may be just as relevant in engineering a resilient SoS. Further investigation should examine metrics—such as flexibility, interoperability, degree of system order, etc.—for their correlation with resilience.
- Extensions to SoS Element Models. The Incubator study treated each system in the SoS as a simple network element that transforms input flows to output flows (flows being information, energy, etc.). An important extension would be to consider richer behavior for these elements. This would enable SoS engineers to consider a number of relevant factors that are not captured in the current mathematical framework. For example, whereas the Incubator project assumed SoS elements either operated normally or were completely disabled, modeling extensions would allow consideration of reduced element performance.
- Formulation of a Decision Support Tool. Incubator project results suggest the ecology-inspired mathematical framework can support decision making in the development of resilient engineered SoS. Future work would result in a prototype tool for supporting such decisions. Such a tool would enable filtering of demonstrably poor SoS architectures (e.g., those outside of the window of vitality), prioritizing the remaining architectures and investigating tradeoffs with other SoS considerations.

7.6.2 Validating SoS Application Studies

The SERC Incubator project focused on a notional surveillance SoS problem. This provided useful insights, but further application studies are needed to validate specific hypotheses and build confidence in a practical SoS engineering methodology. Future studies would be feature larger, more complex SoS architectures and mission scenarios. They also would consider multiple types of SoS problems. Specific trends and claims in need of empirical validation include the following:

- Location of window of vitality. Incubator results indicate that this shifts based on threat severity and we believe there are more fundamental mathematical explanations for this behavior. However, it is important to verify this trend on other types of SoS problems.
- Verification of additional figures of merit. As noted previously, a number of network metrics may be relevant for forecasting the resilience of an engineered SoS. Any hypothesized metrics must be validated against empirical data. Desirable metrics are those that correlate strongly with traditional resilience metrics but that are easier to calculate.
- Independence from threat models. A key motivation for the ecology-inspired mathematical framework is that it does not require detailed models for the threats that might disrupt an engineered SoS. Further empirical data validating this assumption is desired.

The generality of the mathematical framework means it can be applied to a number of SoS problems. Examples of potential application studies include the following (not an exhaustive list).

- **SoS to provide intelligence, surveillance and reconnaissance for protection or rescue**
 - Network Actors: available units for surveillance, data exploitation, rescue or attack, commanding wings of these units, and central mission command. Network flows: operational information (e.g. surveillance data) & commands and reports.
 - Network Design Goal: a network with subsystems from various units of the armed forces would require efficient communication lines and the ability to survive external disruptions. Therefore, a balance of distribution and efficiency of operational pathways will be needed.
- **Supply chain network design**
 - Network Actors: all commercial/federal entities involved in procurement of required components, sub-assembly, final assembly, and distribution/transfer of cargo of interest. Network flows: flow of cargo (number of units, Kgs, etc.)
 - Network Design Goal: sufficient distribution of functionality (procurement, assembly, etc.) as well as selection of multiple and compatible producer consumer relationships to ensure supply despite disruption of operations of few actors.
- **SoS architectures for Naval combat, support and/or surveillance operations**
 - Network Actors: multiple sensors, processors, weapons systems, and the primary command. Network flows: operational information (e.g. surveillance data) and commands and reports.
 - Network Design Goal: Sufficient inter-operability between geographically dispersed participating systems such that the complete range of the battlefield is always under observation and allowing for attack against any detected threats.

7.7 REFERENCES

1. Owens, W.A., *The emerging US system-of-systems*. 1996, Institute for National Strategic Studies, National Defense University, Washington DC.
2. White, B.E. *Fostering intra-organizational communication of enterprise systems engineering practices*. in *National Defense Industrial Association (NDIA), 9th Annual Systems Engineering Conference, San Diego CA*. 2006.
3. Maier, M.W., *Architecting principles for systems-of-systems*. 1998. **1**(4): p. 267-284.
4. Pape, L., et al., *A Fuzzy Evaluation method for System of Systems Meta-architectures*. *Procedia Computer Science*, 2013. **16**: p. 245-254.

5. Uday, P. and K. Marais, *Designing Resilient Systems-of-Systems: A Survey of Metrics, Methods, and Challenges*. Systems Engineering, 2015. **18**(5): p. 491-510.
6. Dagli, C.H., et al., *An advanced computational approach to system of systems analysis & architecting using agent-based behavioral model*. 2013.
7. Uday, P. and K. Marais, *Exploiting Stand-in Redundancy to Improve Resilience in a System-of-Systems (SoS)*. Procedia Computer Science, 2013. **16**: p. 532-541.
8. Hosseini, S., K. Barker, and J.E. Ramirez-Marquez, *A review of definitions and measures of system resilience*. Reliability Engineering & System Safety, 2016. **145**: p. 47-61.
9. Uday, P. and K.B. Marais, *Resilience-based System Importance Measures for System-of-Systems*. Procedia Computer Science, 2014. **28**: p. 257-264.
10. Jackson, S. and T.L.J. Ferris, *Resilience principles for engineered systems*. 2013. **16**(2): p. 152-164.
11. Dekker, A.H., *C4ISR, the FINC methodology, and operations in urban terrain*. Journal of Battlefield Technology, 2005. **8**(1): p. 25.
12. Gillespie, S.E., et al. *System of systems architecture feasibility analysis to support tradespace exploration*. in *2017 12th System of Systems Engineering Conference (SoSE)*. 2017.
13. Holling, C.S., *Resilience and Stability of Ecological Systems*. Annual Review of Ecology and Systematics, 1973. **4**: p. 1-23.
14. Fath, B.D., et al., *Ecological network analysis: network construction*. Ecological Modelling, 2007. **208**(1): p. 49-55.
15. Layton, A., B. Bras, and M. Weissburg, *Ecological Principles and Metrics for Improving Material Cycling Structures in Manufacturing Networks*. Journal of Manufacturing Science and Engineering, 2016. **138**(10): p. 101002-1 – 101002-12.
16. Ulanowicz, R.E., *Growth and Development: Ecosystems Phenomenology*. 1986: iUniverse.
17. Ulanowicz, R., et al., *Quantifying sustainability: Resilience, efficiency and the return of information theory*. Vol. 6. 2009. 27-36.
18. Chatterjee, A. and A. Layton, *Bio-Inspired Human Network Design: A Multi-Currency Robustness Metric Inspired By Ecological Network Analysis*, in *ASME IDETC/CIE*, J.S. Linsey and P. Grogan, Editors. 2019: Anaheim, CA.
19. Borrett, S.R. and A.K. Salas, *Evidence for resource homogenization in 50 trophic ecosystem networks*. Ecological Modelling, 2010. **221**(13-14): p. 1710-1716.
20. Ulanowicz, R.E., *The dual nature of ecosystem dynamics*. Ecological Modelling, 2009. **220**(16): p. 1886-1892.
21. Technology, A.F. *JSTARS – Joint Surveillance and Target Attack Radar System*. 10/13/2019]; Available from: <https://www.airforce-technology.com/projects/jstars/>.

8. QUANTIFYING MISSION IMPACT FOR TECHNOLOGY ALTERNATIVES – DR. ERIC WEISEL, OLD DOMINION UNIVERSITY

8.1 PROBLEM STATEMENT, MOTIVATION, AND OBJECTIVES

A persistent challenge for acquisition stakeholders is a method to value technology alternatives against mission impact that meaningfully informs decision-making for the purpose of relating value and cost. Expected value of information theory provides a well-established basis for valuing various forms of information within a decision-theoretic framework. Our approach is to apply this theory as a basis to value technology alternatives for well-specified mission impacts.

Our objective for this effort is to implement an effective, i.e. algorithmic, method to value model-informed alternatives for well-specified objectives. If successful, this approach will establish expected value of information theory as a basis to quantify the model-informed trade-space between cost of technology alternatives and mission effectiveness. We envision this approach as the basis for an enabling technology to optimize modeling decisions within this trade-space.

The method will be implemented in an EVSI-based scoring model and a set of software modules for kill chain analytics. The resulting EVSI-Based Scoring and Kill Chain Analytics components will be integrated into a separately-funded Mission Engineering and Integration Analytics Platform currently under development.

8.2 BASIS FOR THE RESEARCH OPPORTUNITY

Currently there are few, if any, rigorous tools to value technology alternatives in a mission engineering and integration analysis. Kill chain and mission thread analytics are frequently performed by hand using ad hoc analytics. Given the complexity of mission threads involved in meaningful system acquisition planning, rote and ad hoc tools are insufficient for the task. Expected value of information theory is a promising choice to develop rigorous tools to quantify the model-informed trade-space between cost of technology alternatives and mission effectiveness within a mission engineering and integration framework. Further, there is no way to rigorously evaluate investments in models that inform this decision space. Although significant work has been done to this end, most published efforts focus on accuracy as an objective, leaving only subjective measures to assess sufficiency.

Applying expected value of information theory to inform decision-making is an established approach in other fields, such as health economics, where it has been used to assess the need for continued medical research, to reduce uncertainty in healthcare decision-making, and for sensitivity analysis [Winston, 2004] [Ades et al., 2004]. Within this field, studies demonstrate the effectiveness of expected value of information theory to disconnect the decision to accept a new technology given current information from the decision to continue research [Ades et al., 2004]. Claxton [1999] argues health economics decisions of this nature should be based on expected net gain, the difference between Expected Value of Sample Information (EVSI) and total cost.

Chen and Willan [2014] provide a detailed example, including a link to Excel spreadsheets containing calculations, of using EVSI to analyze a new diagnostic test for pulmonary embolism, specifically informing the decision of whether or not there is sufficient evidence to adopt the new diagnostic strategy, and when there is insufficient evidence, informing the design for additional research.

8.3 TECHNICAL APPROACH

The proposed solution applies Bayesian statistical inference to an EVSI decision structure to iteratively exploit simulation and system performance test data in a mathematically rigorous and defensible way. The method builds upon an existing framework for simulation valuation [Weisel, 2013] consisting of two parts: (1) a model to value simulation alternatives based on use—its purpose is to compare alternatives using expected value of sample information; and (2) a structure for relating artifacts in the simulation activity, the process of constructing and validating a simulation for a well-specified use—its purpose is to provide a framework to collect and reason on evidence for suitability in a rigorously defensible way. Two assumptions apply. The first is that data is available in a useful form. The second is that valuing benefits and costs can be achieved sufficiently for demonstration purposes using standard actuarial methods.

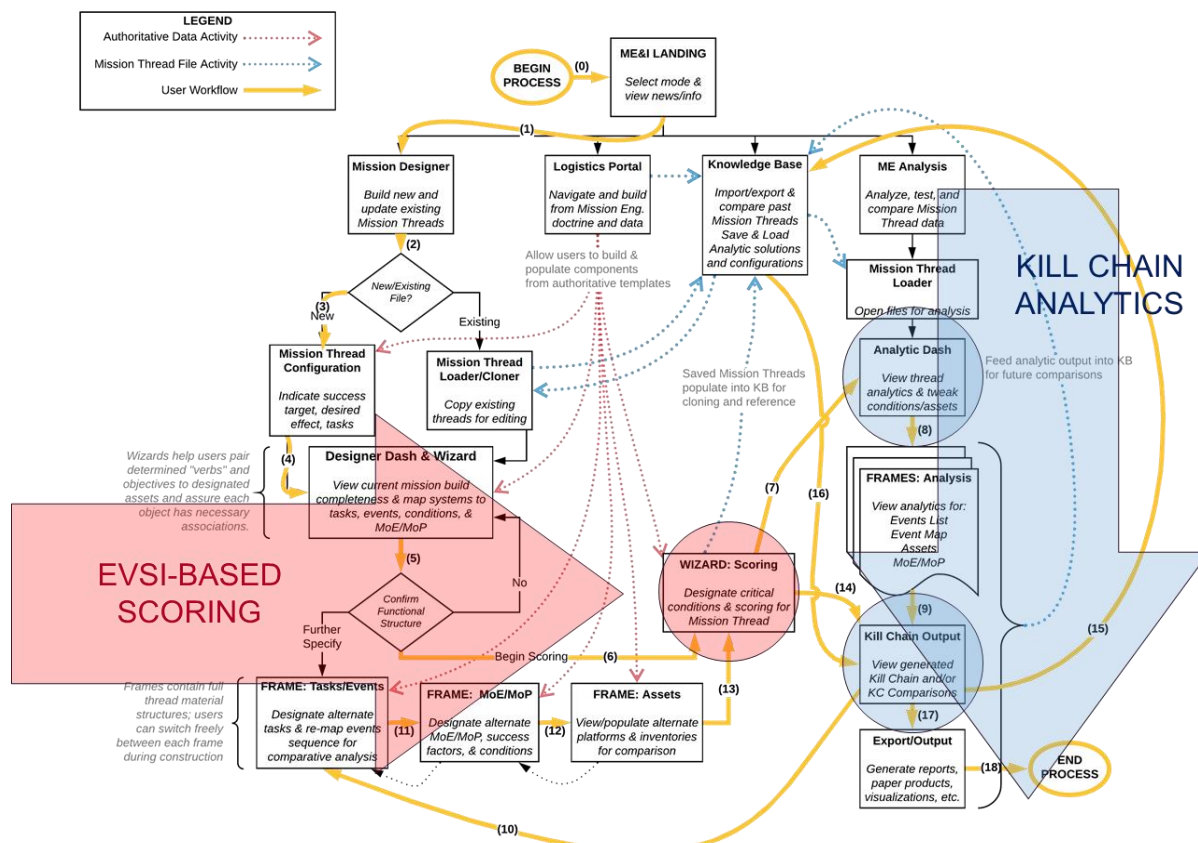


Figure 1. Mission Engineering and Integration Analytics Platform

The primary challenge, which resolves to two key components, is to demonstrate the capability using real-world data. The first key technical challenge is to estimate probabilities within the decision structure directly from real-world simulation and system performance test data. The second is then to apply the resolved decision structure to meaningfully inform smart choices within the trade-space.

The approach calls for three-phased effort focusing on developing prototype software for: (1) EVSI-based scoring model; (2) extension of that model suitable for kill chain analytics; and (3) integration of both into a separately-funded prototype Mission Engineering and Integration Analytics Platform; see Figure 1.



The prototype will be demonstrated in a mechanical design challenge featuring a technology upgrade selection for VMASC's WAM-V autonomous maritime small vessel. The WAM-V autonomous maritime small vessel was developed at VMASC to perform in last year's Office of Naval Research (ONR) Maritime RobotX Challenge. The WAM-V autonomous maritime small vessel is an appropriate demonstration test platform because the challenge requirements are unknown and subject to change as research objectives related to the WAM-V are achieved. A way to rapidly assess technology alternatives to meet challenge requirements would create a competitive advantage for the RobotX team as it prepares for the next competition. The team plans to leverage VMASC's standing range capability, featuring a recent \$400K Commonwealth of Virginia investment in VMASC's simulation experience lab, for most prototype development activities.

8.4 2019 INCUBATOR PROGRAM RESEARCH FUTURE CHALLENGE TASK

The 2019 Incubator Program Research Future Challenge Task developed mathematical analyses to demonstrate the feasibility of the following elements of the technical solution: (1) valuing specific performance-based outcomes; (2) estimating probabilities of outcomes using Bayesian analysis; (3) using subject-matter-expertise to initialize the prior probabilities for the Bayesian analysis; and (4) using expected value or expected utility to meaningfully differentiate the value of model-informed alternatives of varying fidelity for the purpose of assessing technology alternatives based on mission effectiveness and cost.

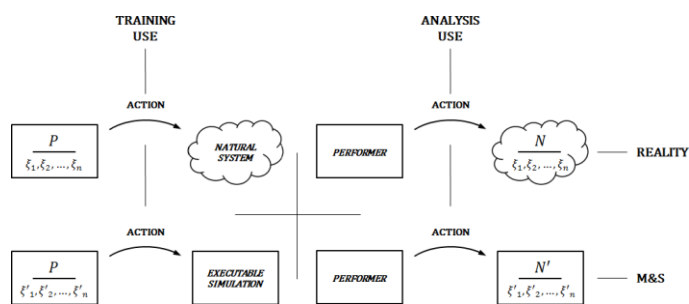


Figure 2. Consequence for Various Uses of Simulation

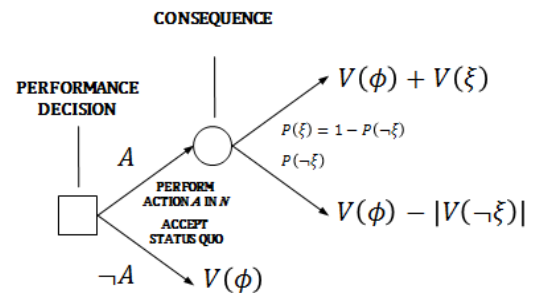


Figure 3. Performance Decision

In order to apply the EVSI framework, or one like it, to value simulation information, we need a model of use. A *performer*, P , is an actor who takes some *action*, A , on a system. The intended result of the action is some *consequence*, the desired outcome, ξ , described with a well-formed statement written as a proposition where $\xi = TRUE$ if some measurable condition or state of the *natural system*, N , or the performer is $TRUE$, $\neg\xi = TRUE$ otherwise. Likewise, $\xi' = TRUE$ if the same condition is $TRUE$ when performing the action on the *synthetic system*, N' . Figure 2 shows this relationship for training and analysis uses. For a training use, the objective will formulate some measurable knowledge or skill observable in the performer. For analysis use, the objective will formulate some measurable condition observable in the system. Now, consider a decision such that a decision-maker, DM , chooses between (1) selecting an action by performer on N with consequence ξ or $\neg\xi$ and probability of occurrence $P(\xi)$ or $P(\neg\xi)$ respectively or (2) rejecting the action, i.e. selecting the complement $\neg A$, and thereby accepting the status quo ϕ with probability of occurrence $P(\phi) = 1$. The *performance decision*, a tuple (N, A, ξ, ϕ) such that the elements of the tuple are related by Figure 3, is the basis for a model for *use*. We experiment with the system/perform the action in simulation with the analytic intent that if the synthetic system responds with the desired condition, ξ' , then the same outcome, ξ , will hold in reality. In both cases, we will seek data to support the case that the desired outcome is reachable with an acceptable risk.

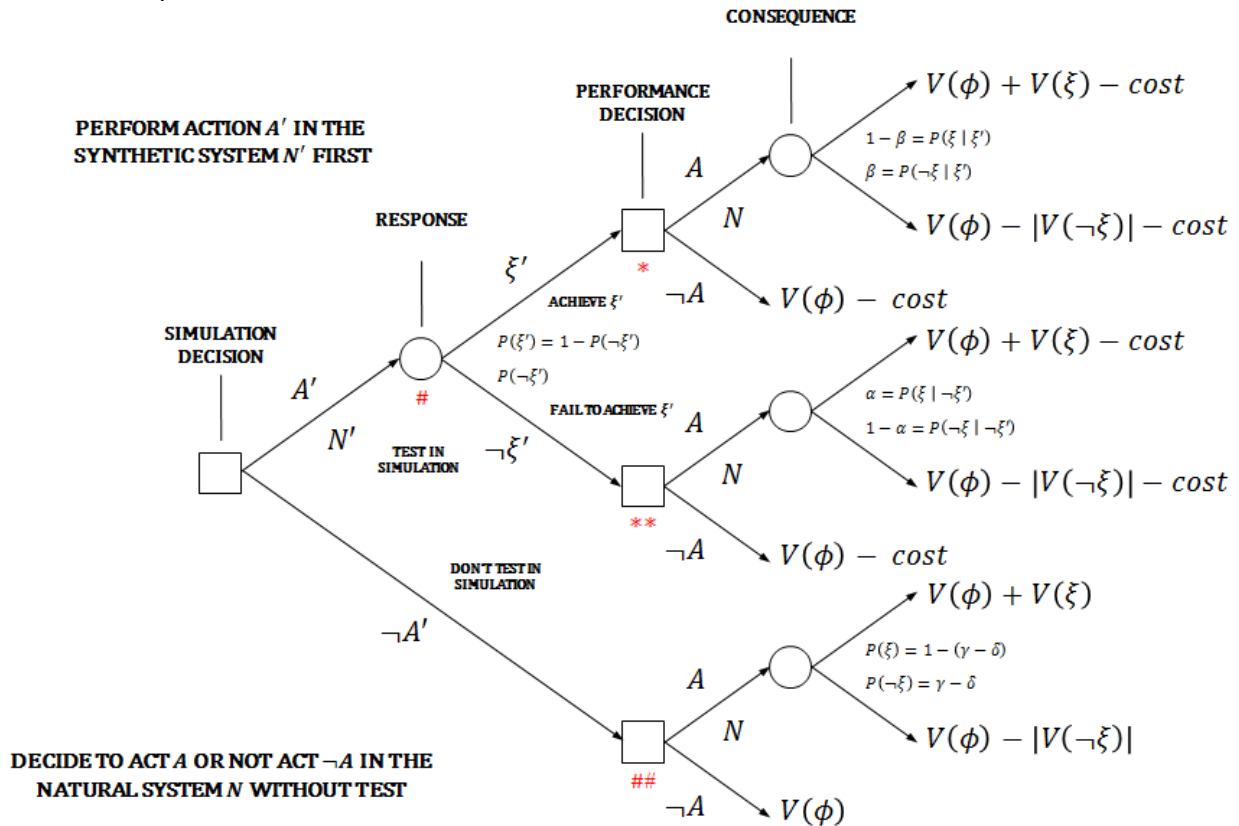


Figure 4. EVSI Analysis to Value Simulation Information

Now, consider a performance decision in which A is to be performed on a system. However, first the decision-maker considers the alternative of first performing A' in simulation. Risk is

structured within the context of an iterative simulation decision; see Figure 4. In general, the decision tree will be algorithmically resolved. Benefits and costs will be estimated using standard actuarial methods, and can include any costs linked to the simulator, such as acquisition, maintenance, operations, and end-of-life costs. EV then quantifies the relative values for the alternatives of performing additional testing in simulation or proceeding directly to performance on the real-world system with no, or no additional, testing in simulation. For example, consider an immersive environment that represents a shipboard space, such as a passageway or crew's quarters, to be used to assess potential locations of shipboard fire-fighting equipment. To test various configurations, the analyst performs a series of experiments testing expert subjects' ability to locate fire-fighting equipment under various casualties in a virtual environment. One of many objectives might be formulated as:

$$\xi = \text{WATCHSTANDER LOCATES AFFF FIRE EXTINGUISHER}$$

$\xi' = \text{TRUE}$ if the performer achieves the objective in the simulator. Our intent in performing the training then is that the performer will likewise achieve the same, $\xi = \text{TRUE}$, in the space on the ship. In this notional scenario the performer, P , is the crew trainee; the natural system, N , is the shipboard space; the synthetic system, N' , is the same space in a virtual environment; the action, A , is to locate the AFFF fire extinguisher in the shipboard space; the action, A' , is to locate the AFFF fire extinguisher in the virtual environment; and the status quo, ϕ , is that the ship is short one watch-stander. See Figure 5. Note that this may be one of many actions and associated objectives analyzed during system design.

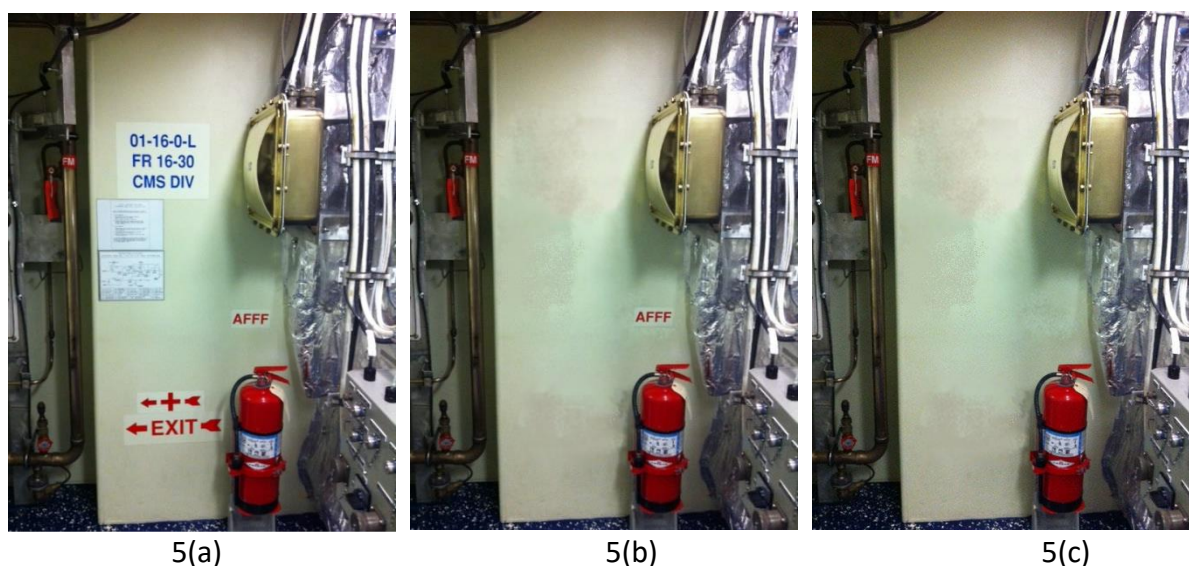


Figure 5. Fidelity Alternatives for Visualization of Bulkhead Onboard LCS-2 Class

A key technical challenge is to develop effective techniques to estimate the probabilities to resolve the decision tree. Indeed, the main assumption in predicting a successful research outcome is that we will be able to defensibly estimate these probabilities. For this effort we will use Bayesian statistics to iteratively update the prior probabilities as new evidence, validation and use data, becomes available. In Bayesian inference, Bayes' rule is used to update the

probability estimate for a *hypothesis* as additional *evidence* about the hypothesis becomes available. The *posterior* probability is calculated from the *prior* probability and the *likelihood* of observing the evidence.

$$P(H|E) = \frac{P(E|H)P(H)}{P(E)}$$

A significant part of the solution to this technical challenge will be to estimate these probabilities using real-world, and expectedly sparse, data.

With the posterior probabilities estimated, EVSI is the difference in Figure 4 decision tree evaluated at # and ##. A risk-neutral DM will choose to invest in simulation if $EVSI > cost$. It is in this sense that EVSI is a measure of value of a simulation alternative for a well-specified use. To demonstrate how the analysis sheds light on valuation, consider the following case, in which the probabilities of Type I and Type II errors are small:

$$EVSI = V(\xi) \left[(1 - \gamma)[P(\xi'|\xi) - P(\xi'|\neg\xi)] - \delta \left(P(\neg\xi'|\xi) + \frac{(1 - \gamma)}{\gamma} P(\neg\xi'|\neg\xi) \right) \right]$$

This equation for EVSI is meaningful beyond a reduction of the algorithm to an equation. EVSI is calculated directly from measurable test data without requiring estimation of posterior probabilities. Further, this equation separates component contributions, such as the contributions made by simulation accuracy and the expected value of the original information, to the overall expected value of the sample information. One benefit is to define conditions that will increase the value of a simulation-based analysis for a set of well-defined analytic objectives. Further, EVSI then forms the value structure to derive an objective function for query-based trials and optimization.

Phase 1—EVSI Based Scoring and Feasibility of Extension to Kill Chain Analytics

In the first Phase, EVSI will be used as a basis for a scoring model intended for application in a Mission Engineering and Integration Analytics Platform. Further, the first Phase will investigate the feasibility of extending the scoring model, through composition, to score composed asset × condition × mission threads in a Kill Chain Analytics module. Phase 1 is designed to demonstrate feasibility by: (1) solving the highest-risk technical challenges first; (2) theoretically bounding technical challenges; and (3) seeking robust heuristic solutions to problems with computational complexity limitations. The Phase 1 effort will:

- Develop a method for persistent data collection from simulation analytics
- Develop an EVSI-based model for asset × condition × mission scores
- Assess the feasibility of extending the EVSI-based model to kill chains and mission threads
- Develop an EVSI-based scoring TRL-4 software component
- Perform a technology demonstration in a laboratory environment

Software development will be performed in four spirals, nominally with three four-week sprints per spiral. Each spiral will conclude with a quarterly technology demonstration.

Task 1—EVSI-Based Scoring Model

Literature Survey. Conduct a thorough literature survey with respect to the sciences and technologies applicable to EVSI-based scoring. Identify key mathematical structures and theorems, open questions, case studies, technical challenges, and risk. Evaluate the state-of-the-art with respect to EVSI. Document findings. *Measurable milestones: Completion of draft.*

Basic Research and Research Integration. Develop research solutions to key technology challenges. Perform fundamental research leading to new science enabling EVSI-based scoring of: (1) Technologies and comparisons of technologies; (2) Asset $\times$ Condition $\times$ Mission tuples; and (3) Kill chains and mission threads. Develop research solutions to enabling technologies by software component. Evaluate options for persistent data collection from simulation analytics. Perform research integration. Document findings. *Measurable milestones: Completion of drafts.*

Feasibility Study. Perform detailed analyses focusing on feasibility of Extending the EVSI-based model to kill chains and mission threads. Document findings. *Measurable milestones: Delivery.*

Deliverables: (1) Feasibility Demonstration; (2) Interim Report.

Task 2—EVSI-Based Model Component Prototype

Software Concept, Architecture, and Planning. Develop organizational deployment and use concept. Perform software system requirements analysis. Develop software architecture concept. Develop detailed UML implementation. Perform detailed concept and architecture review and identify areas of technical risk. Prepare Software Concept and Architecture Design Document. Prepare software development environment. Establish configuration and change management process(es). Establish project management process(es). *Measurable milestones: Completion of software concept work product.*

Software Component Prototype Development. Develop executable architecture and prototype algorithms for each identified technical risk. Develop an EVSI-based model component prototype. Prepare Software Design Document. Perform detailed design review. *Measurable milestones: Completion of detailed design work product.*

Software Integration. Develop TRL 4 prototype software. Perform executable architecture and prototype testing in a laboratory environment. *Measurable milestones: Completion of prototype software.*

Software Test and Deployment. Perform software unit testing. Perform test review. Perform Spiral 1, 2, 3, 4 technology demonstration. Prepare Component Test Document. Deploy prototype software. *Measurable milestones: Delivery.*

Deliverables: Prototype Software (at each spiral).

Task 3—Mechanical Design Challenge Problem

Mechanical Design Challenge. Develop a Mechanical Design Challenge problem suitable to provide a convincing feasibility argument when implementing the Phase 3 integrated prototype on a relevant technology selection problem. *Measurable milestones: Completion of draft.*

Deliverables: (1) Mechanical Design Challenge Problem Description Document; (2) Interim Report.

Phase 2—Kill Chain Analytics

In the second Phase the Kill Chain Analytics module will be developed with the following features:

- Develop a method to compose a set of asset × condition × mission scores
- Extend the scoring model to kill chains and mission threads
- Develop a kill chain analytics TRL-4 software component
- Perform a technology demonstration in a laboratory environment

Software development will be performed in four spirals, nominally with three four-week sprints per spiral. Each spiral will conclude with a quarterly technology demonstration.

Task 1—Extend EVSI-Based Scoring Model

Basic Research and Research Integration. Continue research solutions to enabling technologies by software component. Evaluate options for persistent data collection from simulation analytics. Perform research integration. Document findings. *Measurable milestones: Completion of drafts.*

Deliverables: Interim Report

Task 2—Kill Chain Analytics Component Prototype

Software Concept, Architecture, and Planning. Refine software concept. Refine software system requirements analysis. Refine detailed UML implementation. Refine software development environment. Perform configuration and change management process(es). Perform project management process(es). *Measurable milestones: Update of software concept work product.*

Software Component Prototype Development. Develop executable architecture and prototype algorithms for each identified technical risk. Develop a Kill Chain Analytics component prototype. Prepare Software Design Document. Perform detailed design review. *Measurable milestones: Completion of detailed design work product.*

Software Integration. Develop TRL 4 prototype software. Perform executable architecture and prototype testing in a laboratory environment. *Measurable milestones: Completion of prototype software.*

Software Test and Deployment. Perform software unit testing. Perform test review. Perform Spiral 1, 2, 3, 4 technology demonstration. Prepare Component Test Document. Deploy prototype software. *Measurable milestones: Delivery.*

Deliverables: Prototype Software (at each spiral).

Task 3—Mechanical Design Challenge Problem

Mechanical Design Challenge. Develop a Mechanical Design Challenge demonstration plan suitable to provide a convincing feasibility argument when implementing the Phase 3 integrated prototype on a relevant technology selection problem. *Measurable milestones: Completion of draft.*

Deliverables: (1) Mechanical Design Challenge Problem Planning Document; (2) Interim Report.

Phase 3—Integration of EVSI-Based Scoring and Kill Chain Analytics Components in Mission Engineering and Integration Analytics Platform

Finally, the third Phase will integrate the software components developed in the first two Phases into the Mission Engineering and Integration Analytics Platform. Development of the software prototype will be a significant software development effort, warranting careful attention to employing effective software engineering practices. We will utilize an agile software development process modeled on the Rational Unified Process for software engineering to ensure reusable software from the component to integrated prototype level. Software development will be performed in four spirals, nominally with three four-week sprints per spiral. Each spiral will conclude with a quarterly technology demonstration. The Phase 3 effort will have the following features:

- Integrate EVSI-based scoring and kill chain analytics software component into TRL 6 prototype
- Perform a technology demonstration of the TRL 6 prototype in a simulated operational environment

Task 1—Integrated EVSI-Based Scoring and Kill Chain Analytics Software Prototype

Software Concept, Architecture, and Planning. Refine software concept. Refine software system requirements analysis. Refine detailed UML implementation. Refine software development environment. Perform configuration and change management process(es). Perform project management process(es). *Measurable milestones: Completion of software concept work product.*

Software Integration. Develop TRL 6 integrated prototype software. Perform executable architecture and prototype testing. *Measurable milestones: Completion of prototype software.*

Software Test and Deployment. Perform integrated software unit, unit integration, and system testing. Perform test review. Perform Spiral 1, 2, 3 technology demonstration. Prepare Integrated Test Document. Prepare Software Documentation. Deploy prototype software. *Measurable milestones: (1) Completion of integrated test work product; (2) Delivery.*

Deliverables: (1) Prototype Software (at each spiral); (2) Software Documentation; (3) Integrated Software Demonstration; (4) Final Report.

Task 2—Mechanical Design Challenge Demonstration

Mechanical Design Challenge. Perform integrated software unit, unit integration, and system testing using the Mechanical Design Challenge developed in Phases 1 and 2. Identify, select, and install technology alternatives. Demonstrate the assembled mechanical design system achieves well-specified mission objectives or functions to the Mechanical Design Challenge developed in Phases 1 and 2. Perform Mechanical Design Challenge review. *Measurable milestones: Delivery.*

Deliverables: (1) Mechanical Design Challenge Demonstration; (2) Final Report.

8.5 RESEARCH TEAM QUALIFICATIONS — PLANNING FOR SUCCESS

The Virginia Modeling, Analysis, and Simulation Center (VMASC) is an applied research center of Old Dominion University (ODU) focusing on innovation, workforce development, and industry ecosystem engagement programs leading to digital transformation. VMASC’s staff of over 50 research faculty, scientists, and support professionals perform scientific research, develop computational models, and create information-to-insight and digital engineering solutions in the areas of:

- Cybersecurity and critical infrastructure systems
- Digital shipbuilding
- Spaceflight and autonomy
- Digital health and health equity
- Policy and decision-making
- Mission engineering and integration
- Warfighter performance and readiness

Our faculty are not traditional academic faculty, rather they are dedicated researchers, generally free from teaching responsibilities, with full-time efforts assigned to research duties. Although our projects are staffed primarily with research professionals, last year VMASC funded 34 students across 5 departments in the Sciences, Engineering and, Arts—we envision significant student engagement in the proposed project.

8.6 RELEVANCE OF THE RESEARCH

A solution will provide significant cost-savings to the government by quantifying the decision space spanning technology alternatives against mission impact in a way that enables a sensible tradeoff between cost and the fidelity of the models informing the decision space. Further, a successful research outcome will: (1) provide a rigorous mathematical basis for design of experiments for testing model-based alternatives; (2) demonstrate the use of expert opinion as initial evidence via the Bayesian priors; (3) formalize growing confidence in model-informed results, even when initial probabilities are difficult to quantify, i.e. subjective initial priors become less important as more testing, or more use, is completed; and (5) develop techniques to quantify

the value of mission effectiveness using familiar financial metrics such as Expected Value of Sample Information and Return on Investment.

8.7 RISK ASSESSMENT AND MITIGATION PLAN

The project has both technical and programmatic risks. ODU has selected a team of highly qualified performers and engineered a process-driven research plan that minimizes risk. Technical risk is minimized by focusing the highest-risk technical challenges, specifically solution performance with sparse data sets, in Phase 1. This program is fast-paced—a key component of managing risk is a compact research and development team, unburdened by long-distance collaboration. All proposed performers are VMASC research staff. The most significant management effort will support software development. Software development risks are minimized by a formal process for software development. Schedule and cost risks are minimized by (1) extensive experience of the research team; (2) rigorous application of a formal research process; and (3) disciplined project management best practices.

8.8 PROJECT COST, SCHEDULE AND MILESTONES

A rough-order-of-magnitude scope for the effort is \$770K over 3 years. In advance of the final year, VMASC will seek additional matching funding of \$750K from commercialization or government program partners to complete the funding needed for year 3 integration tasks. Project evaluation and measurement will be achieved by a series of technology demonstrations coinciding with the completion of each one-year phase of prototype development. The project evaluation will culminate in a final technology demonstration of the integrated prototype in a simulated operational environment. Achievement of internal milestones will be documented in agile work products.

Phase 1—EVSI Based Scoring and Feasibility of Extension to Kill Chain Analytics

- Feasibility Demonstration
- Prototype software (at each spiral)
- Mechanical Design Challenge Problem Description
- Interim Report

Phase 2—Kill Chain Analytics

- Prototype software (at each spiral)
- Mechanical Design Challenge Problem Plan
- Interim Report

Phase 3—Integration of EVSI-Based Scoring and Kill Chain Analytics Components in Mission Engineering and Integration Analytics Platform

- Prototype Software (at each spiral)
- Software Documentation

- Integrated Software Demonstration
- Mechanical Design Challenge Demonstration
- Final Report

Progress will be documented in interim and final reports by Phase in the form of: (1) technical manuscripts of publishable quality and suitable for publication in a peer-reviewed journal documenting effort to date; and (2) programmatic manuscripts containing financial data and other information not suitable for publication but appropriate for program documentation.

8.9 REFERENCES

Ades, A. E., G. Lu and K. Claxton. 2004. Expected Value of Sample Information Calculations in Medical Decision Modeling. *Medical Decision Making* 24, pp. 207-227.

Chen, M. H., and A. R. Willan. 2014. Value of Information Methods for Assessing a New Diagnostic Test. *Statistics in Medicine* 33, pp. 1801-1815.

Claxton, K. 1999. The Irrelevance of Inference: A Decision-Making Approach to the Stochastic Evaluation of Health Care Technologies. *Journal of Health Economics* 18, pp. 341–364.

Weisel, E. W. and M. D. Petty. 2013. A Bayesian Approach to Assessing Expected Utility in the Simulation Decision. *Proceedings of the 2013 Summer Simulation Multi-Conference*, Toronto, ON, Canada, July 7-10, 2013.

Weisel, E. W. 2013. *Expected Value of Sample Information as a Measure of Return-On-Investment in Simulation*, Engineering and Applied Science, MODSIM World Conference, Hampton VA, May 2, 2013.

Weisel, E. W. 2012. A Decision-Theoretic Approach to Defining Use for Computer Simulation. *Proceedings of the 2012 Autumn Simulation Multi-Conference*, San Diego, CA, October 28-31, 2012.

Winston, W. L. 2004. *Operations Research*. Thompson Learning, Belmont, CA, pp. 758-763.