Simulation

Methodology

Simulation: Transactions of the Society for Modeling and Simulation International
2023, Vol. 99(2) 201–220

DOI: 10.1177/00375497221114861
journals.sagepub.com/home/sim
SAGE SCS

The application of applied category theory to quantify mission success

Robert K Garrett[1], James P Fairbanks[2], Margaret L Loper[1] and James D Moreland[3]

Abstract

Mission engineering is the quantification of the effects applied by a system of systems to achieve measurable desired results. The execution of the mission is defined by a mission thread; that is the sequence of actions/processes executed by elemental systems. The domain of complex missions has been described as "wicked" because traditional military and space program-based systems engineering practices fail due to a lack of discrete phases, a dependence on context, and the non-uniqueness of a "good-enough" mission thread. Wicked problems also tend to be unstructured (non-hierarchical) with no centralized control and do not lend themselves to linear step-by-step processes. Wicked problems are inherently uncertain leading to the broader issue of trust across a mission knowledge base, and any mission level analyses. Wicked problems are also characterized and challenged by combinatoric complexity. Mission success is primarily driven by the interrelationships between systems and not just by the individual systems themselves. The success of mission engineering will require an iterative approach of modeling, simulation, and analysis resulting in a continuous reduction in uncertainty and refinement in the topology of the mission thread. OODA-based decomposition of mission threads focused on Boyd's Orient function provides focus on system interrelationships. Trust provides decision-maker confidence in the results that has proved elusive to traditional validation approaches. The approach described in this paper is based upon using applied category theory as a basis for building mission threads, mission models, data stores, and integrating simulation ensembles.

Keywords

Modeling, simulation, mission engineering, system of systems, trust, validation, applied category theory, OODA, complexity

1. Introduction

Mission engineering (ME) is often defined as the "planning, analyzing, organizing, and integrating of current and emerging operational and system capabilities to achieve desired warfighting mission effects."[1] This definition is focused on "management-governed approaches" that consider existing operational platforms and the engineering activities required to purposefully support the mission.[2] In order to get to a computational definition of ME, we first consider that missions operate upon a system of systems (SoS) composed of both systems and system interactions. The execution of the mission is defined by mission threads; that is, the sequence of actions/processes executed by elemental systems. Typically, there are many plausible mission threads that can be executed from a given set of systems. These mission threads may not be independent and can share common sub-paths. Therefore, thinking about ME from a computational perspective leads us to redefine ME as the quantification of the effects applied by a SoS to achieve measurable desired results.

Missions can be complex if they are constructed with many moving parts that originate from sub-optimized business rules. A complex mission can utilize 10s to 1000s of individual systems. For example, consider a ballistic missile

[1]Georgia Tech Research Institute, Information and Communications Laboratory, USA
[2]Computer and Information Science and Engineering Department, University of Florida, USA
[3]OUSD (Acquisition & Sustainment) / ASD (Acquisition), Former Executive Director, Mission Engineering and Integration, United States Department of Defense, USA

Corresponding author:
Margaret L Loper, Georgia Tech Research Institute, Information and Communications Laboratory, 756 W Peachtree St NW, Atlanta, GA 30308, USA.
Email: margaret.loper@gtri.gatech.edu

defense system composed of 100s of satellites, land- and sea-based radars, command and control systems, and a range of missile intercept systems (https://www.mda.mil/system/system.html). The domain of highly complex missions has been described as "wicked"[3] because traditional military and space program-based systems engineering (SE) practices fail due to a lack of discrete phases, a dependence on context, and the non-uniqueness of a "good-enough" mission thread. Wicked problems also tend to be unstructured with no centralized control, or simple hierarchical structure, and do not lend themselves to linear step-by-step processes. Wicked problems are inherently uncertain due to both combinatoric complexity and the epistemic uncertainty about context. This uncertainty leads to the broader issue of trust across a mission knowledge base, and any mission level analyses.[4,5]

Mission analysis is challenged by a lack of quantitative, testable mission architectures, system data including models and simulations, defined mission threads, and actionable context. Validation of mission analyses is challenging at best by the intractable problem of getting sufficient bounding test data consistent with both mission threads and context. Mission analyses should be conducted on a representative set of bounding mission threads to interrogate the effects of uncertainty and complexity. A mission can be constrained or benefited by dynamic/evolving contexts, including things like local and geo-politics, weather, availability of assets, locational information, and so on. The dynamic interplay between the SoS and the environment precludes a single best answer, although there may be many good enough answers. In other words, there are most likely multiple plausible mission threads to achieve a given mission success measure. Quantifying mission success with the current system-centric toolbox has been problematic due to a combination of complexity, context, data management, data analytics, and ever-challenging validation of models and simulations.

The ME domain is heavily reliant on a modeling and simulation (M&S) toolbox to predict mission success. This is due to the exorbitant costs and operational system availability of conducting in situ testing and evaluation of Department of Defense (DoD) mission threads. Traditional system evaluation M&S tools struggle to characterize performance at the mission level. Typically, these tools are structured, discrete-event simulations that are successfully used at the system level, but fail for many reasons when applied to an SoS within a mission context.[6,7] For example, simulations have disparate structured databases which focus communication on data processing and decision processes, with a prescription to a pre-defined sequence diagram. Resolving these issues is time-consuming and expensive, limiting the number of mission threads that can be analyzed. Kinder has proposed moving away from a single M&S approach that attempts to provide a true/false answer for mission success. To evaluate an SoS, he embraces a toolbox of structured and unstructured M&S approaches providing various quantitative perspectives on acceptable mission success criteria[8] thus providing richer insight into the mission context. The nature of a complex mission will require an iterative approach using multiple tools resulting in a continuous reduction in uncertainty, an increase in Trust, and refinement in the topology of the mission thread. Trust in all aspects of the mission must be measured as a basis for decisions given the challenges of validation.

A key premise in this paper is that mission success is dominated by the inter-system relationships, or interstitial space,[9] that space between where integration resides. This paper proposes a multi-faceted and inherently unstructured approach to better quantify mission success focused on quantification of the interstitials, while exploiting current M&S tools. The intent of this approach is not necessarily to model and simulate the totality of a mission, but to provide appropriate levels of abstraction and insight where needed within a mission due to a lack of empirical data, or a change in context. This enables foundational mission data for future mission analyses. Through realistic examples, the creation of a multi-layered, multi-dimensional mission model, and event chains will be demonstrated. The starting point will be a U.S. DoD proposed ME approach. This process will then be rigorously expanded and applied to a generic non-DoD mission within a surrogate example for general practice understanding. The surrogate example will utilize a city neighborhood involving people transiting about the neighborhood and participating in the functions of education and work, with an underlying behavior of cheating. The focus of the effort is on the establishment of a rigorous mission schema along with an approach to mission functional decomposition based on the Observe, Orient, Decide, and Act (OODA) loop.[10] In the OODA construct, the Orient function is closely aligned with the interstitials while Observe, Decide, and Act tend to be entity functions. Finally, an overarching approach to quantify trust will be presented to address problems with M&S validation at the mission level.

The remainder of this paper is organized as follows. Section 2 defines ME and some key limitations in the current SE process. Section 3 expands on the idea of ME by discussing mission threads and the importance of the Orient function. In Section 4, we introduce applied category theory (ACT), a mathematical paradigm for constructing multidomain missions to enable integration of SoS models. An iterative approach to quantitatively evaluating the ability of an SoS to meet mission goals is presented in Section 5. The concepts from Sections 3–5 are illustrated in an example using a city and students in Section 6. We conclude the paper with a discussion of trust and its relationship to validation of SoS ME in Section 7, and then present our conclusions.

2. The DoD ME process

The authors are engineering practitioners who have spent their careers applying M&S tools to various problems in and around the US DoD. Quantifying the performance of

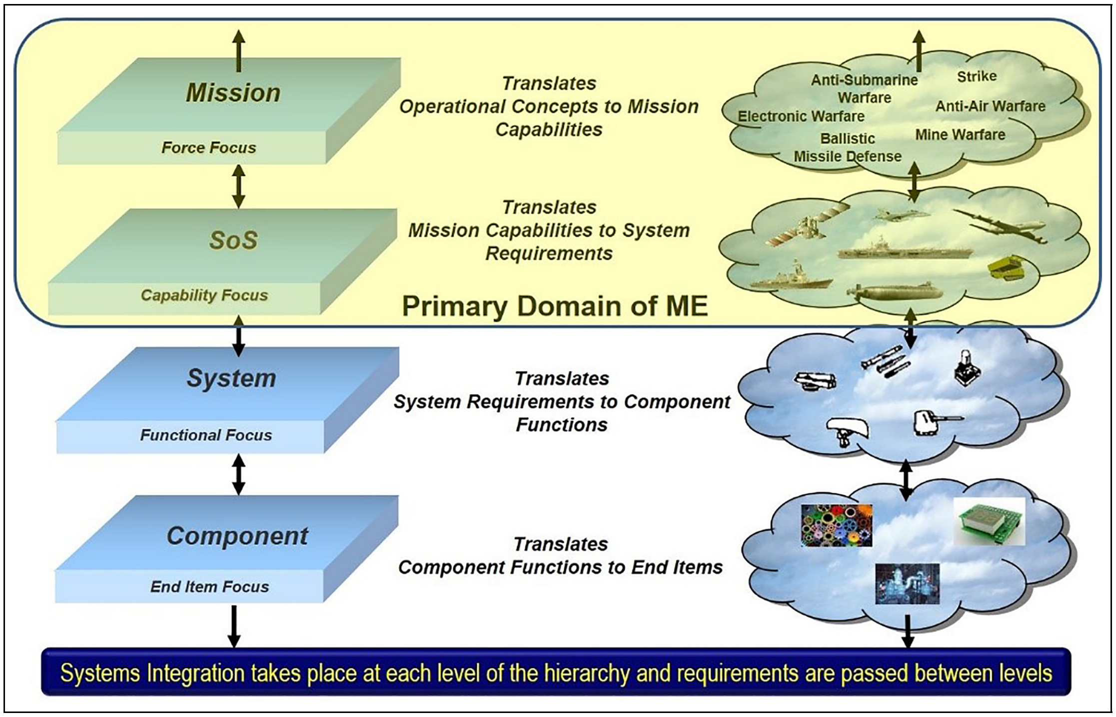

Figure 1. DoD ME hierarchy.

an SoS within the context of a complex mission and mission environment using M&S is the common ground for their collaboration. Their efforts presented here were foundational in the creation of the recent DoD focus on ME and integration.

2.1. Background and limitations

In today's DoD acquisition environment, programs are matured independently and SoS integration occurs when delivered to the field rather than during early initial development. Figure 1 illustrates the various domains of SE applications within the DoD by showing the different levels and interactions using the SE hierarchy. The acquisition workforce has been primarily focused on refining the process and content of the system and component domains over the last few decades; however, this same focus is now required at the mission and SoS levels to address integration and interoperability shortfalls across a multi-domain force. Historically M&S has connected just two layers of this hierarchy, usually only in one direction. Given the complexity of current and future weapons, the engineering community must ensure the mission and SoS levels are thoroughly and rigorously documented to drive appropriate design at all levels.

The complex and now highly integrated machines of warfare continue to evolve enabling higher precision, more effective power projection and safer defensive postures for the DoD services. The interconnectedness of our own social fabric is finding its way into our ships, aircraft, submarines, networks, space assets, tanks, and the very weapons they deliver. Extensibility of the M&S toolbox along with mission architectures and mission threads is key as combinatoric complexity is ever increasing. Determining the right investments for the development of space assets and the protection of positioning, navigation, and timing applications is extremely complex requiring this systematic approach to identify all the major interfaces and intersection points among many systems/platforms in a multi-domain, multi-functional environment. Understanding SoS executing along mission threads within defined context at this level of detail provides the opportunity to reduce the risk of fielding broken mission capabilities as one looks across warfighting domains and individual DoD services. Validation of mission level analyses has been problematic. Traditional verification and validation (V&V) processes tend to rely on comparisons to test data, but mission level data are rarely available. In addition, this approach will allow one to pay attention to resilient alternatives to executing mission threads while reducing unnecessary redundancy within and across DoD services. Key requirements for decision makers from ME analyses must include transparency and sharing of data and models, a collaborative environment to install simulation and software tools, and curation of trustworthy data.[11]

The DoD developed and started integrating an ME approach in 2014 to address the shortcomings of SE and SoS engineering (SoSE) processes to quantify mission success. The state of art in ME when using SE and SoSE tools was overly reliant on language-based artifacts is system centric and frequently context independent. As described above, the authors see the key limitations in the SE and SoSE processes when applied to ME as

- The inability of models and simulations to work collaboratively from component to system to SoS to the mission level, and back down;
- The lack of robust and extensible mission architectures that are math-based and testable;
- The inability to quantify the interrelationships between systems along defined mission threads as a function of context; and
- V&V techniques developed for component and system M&S are problematic at the mission level.

Today's government research organizations are facing increased competition not just from adversaries but with neutrals and Allies in a world where access to technology is ubiquitous. Thus, the DoD must not only equip its fighting force to defeat what is anticipated but equip the force to defeat an adaptable future adversary in a wide variety of environments. The fundamental question then is how to design a research capability that can not only operate but also excel in this environment. Critical to the success of this objective is the need to strengthen the connection between the research community and those responsible for fielding an effective and adaptable warfighting force. This relationship needs to be built upon a common understanding of mission thread execution and tied to the military decision-making process to reinforce both its relevancy and the appropriate funding levels. To strengthen that bridge, an ME approach can be used to link research and innovation objectives to warfighting capabilities.

2.2. Event chains and mission graphs

Our approach to ME is a systematic, quantifiable, and iterative approach combining the structure of SE and the tactical insights of operational planning to achieve mission wholeness. The ME approach begins as a language-based analysis with the collection of mission information. These data are retrieved from doctrine and policy, and include an initial mission thread(s) defining context and the operational mission environment. The findings are captured in "effects/kill chains" to clearly identify operational needs based on the way one plans to execute mission threads that are captured in operational plans and contingency plans. The effects/kill chains (referred to as event chains throughout the paper) are composed of a set of tasks (e.g., track threat, detect hostile intent, and neutralize threat) and represent a path in a graph.

Each node/system in the path is subjectively ranked as red/yellow/green based on defined success criteria. In this analysis, the edges are only implicitly addressed. As the analysis proceeds, tactical systems replace conceptual nodes in the chains. Event chains are then reconfigured to maximize *green* capability. Based on these analyses, the mission threads are reconfigured and the analysis repeated as goals evolve. This approach is unique based on the complex integration of models, mission threads (scenarios), simulation, optimization, and uncertainty upon a common graphical framework. The graphical framework begins as a conceptual model for a mission that evolves into a data schema, simulation architecture, extensible database, and data analytics platform.

While complexity is frequently used in describing SoS, it is rarely quantified.[12] To address this issue, we introduce a mission graph as a key concept of ME—an overarching, executable, and extensible datastore containing the entirety of a mission-specific toolbox. The contents of the toolbox contain relevant M&S software; relevant system, SoS, and mission data including models, simulation and test results; and system, SoS, and mission-specific contextual data to include geography, weather, policy, statutes, and doctrine. With a mission graph, complexity can then be defined as a measure of extent based on an ordered triple (n, e, p) where n is the number of systems/nodes, e is the number of system–system interaction/edges, and p is the number of defined paths, or mission threads in a mission graph. In the mission graph, the number of edges can approach $O(n^2)$ (n squared) and the maximum number of paths can approach n! (n factorial).[13] Each node and edge in the mission graph contains both structured and unstructured information to include metadata, functional behaviors, and empirical and virtual data.[14]

The Guidebook[15] uses a simplistic, notional Air Warfare example to demonstrate the event chain process. The scored event chain, using nodes, edges, and paths (n, e, p), is shown as a string diagram in Figure 2. The colored circles are the nodes, the gray lines connecting the nodes are edges, and the dotted edges indicate alternate paths where multiple systems (nodes) come into play. The paper will build upon this example.

3. Mission thread, OODA, and the orient function

Many military-based event chains are deliberately system-centric in their construction to fit within a required SE process. Evaluation of communications and messaging beyond their system boundaries (particularly in virtual and constructive testing) are usually only implicitly considered. System level event chains also tend to be system unique in lexicon with a detailed level of abstraction that is unnecessary for mission analysis. This provides new challenges when aggregating diverse systems into a mission-based SoS. An SoS/mission event framework with

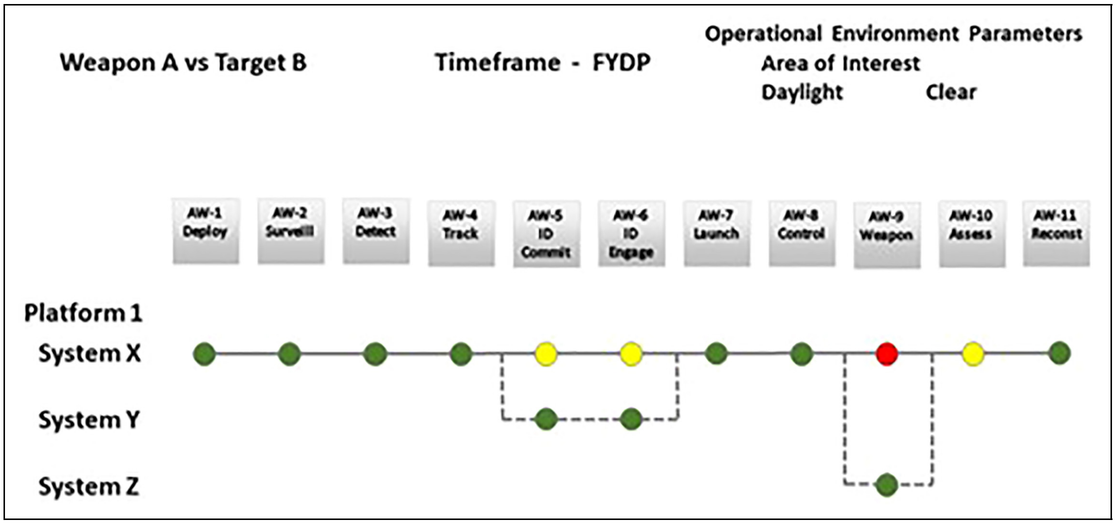

Figure 2. Scored event chain applied to an SoS consisting of three systems has complexity (14, 15, 4).

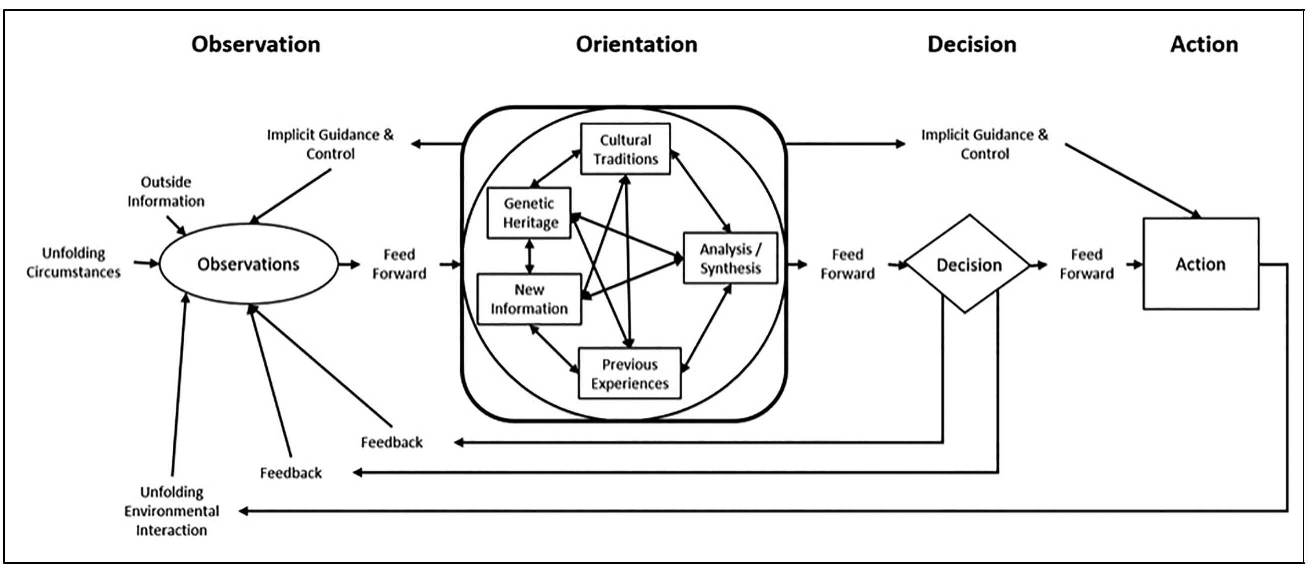

Figure 3. John Boyd's OODA Loop.

an appropriate level of abstraction that explicitly represents communication and data processing across the SoS, and provides for contextual awareness is desirable.

In the Air Warfare example in Figure 2, the functions of information technology (e.g., communications, information technology, data analytics, and artificial intelligence) are represented as edges in the graph. Unfortunately, these edges only assume that a relationship exists, the details of that relationship are not defined. For example, how do systems X, Y, and Z communicate, are satellites and non-organic command and control involved? These edges are the interstitials,[9] which are the domain of integration and interoperability across the mission. The interstitials thus play a dominate role in mission success and need to be explicitly represented in the event chain. The process to address the interstitials is based on Boyd's[10] OODA loop and shown in Figure 3.

Where Observe, Decide, and Act are system functions, Boyd defined Orient as a multi-faceted and iterative hub between them. It is Orient that is suited to represent the interstitial functions. Boyd defined the first five sub-functions to Orient, contextual awareness and external communications were not considered. The last two Orient sub-functions (New Information and Genetic Heritage) reduce uncertainty, and enable better decisions and informed actions.[16]

- *New Information*, *Previous Experience*, and *Analysis/Synthesis* are straight forward involving data processing and extend readily to ME.

Table 1. Translation of Boyd's Orient function to a suitable SoS construct for ME.

Boyd 1987	ME 2020
Cultural Traditions	Parse the physical and natural environment relevant to the moment
Genetic Heritage	Parse the human environment relevant to the moment
New Information	Parse and analyze the Observe/sensor network data, and update the knowledgebase
Previous Experience	Mine the historical data, LVC, etc., update the knowledgebase
Analysis/Synthesis	Re-calculate "real-time" ability to meet mission goal and effects chain goal
Communication	Communicate

LVC: live, virtual, and constructive.

- *Genetic Heritage* and *Cultural Conditions* involve inference, and are a means of assessing the local environment including social context within the mission environment.

A sixth sub-function, communication, is added here to explicitly address messaging across the SoS. The communication sub-function is more than having the means to communicate (e.g., the pipe); it includes what flows on the pipe (syntax and semantics, quality, trustworthiness, timeliness) and the unique needs of the two systems (nodes) that are connected by the pipe (edge). Thus, communication along with the other five Orient functions is part of the Interstitial Space, a foundational characteristic between every system/subsystem in the mission. Essentially, we are ensuring syntactic and semantic interoperability. The mapping from the Boyd fighter pilot perspective to an SoS mission perspective is shown in Table 1. "Relevant to the moment" with respect to the environment is subjective and is tested in mission thread architecture and subsequent simulation. It is another source of uncertainty.

Figure 4 maps the Air Warfare events from Figure 2 to OODA functions. Interestingly, there are no Orient steps to enable data flows and communication. To create an event chain that represents the networked SoS, these OODA functions are then explicitly interspersed with Orient functions. Figure 5 shows plausible Observe and Decide event chains, with the Orient function explicitly represented where Orient is one of the first four sub-functions in Table 1. In these event chains, the *communication sub-function* is represented as a directed edge. The loops about the Orient functions represent iterative processing and can add significant complexity. It is plausible that repetitive looping between multiple Orient functions can greatly increase combinatoric complexity. These graphs are not unique solutions to the event chain but represent plausible paths within a mission thread. Presenting event chains and mission threads as OODA-based graphs will set the stage for subsequent quantitative analyses.

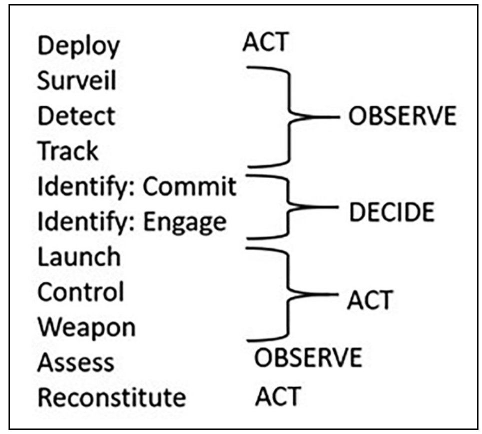

Figure 4. Air Warfare event chain mapped to the OODA functions.

4. Applied Category Theory

While knowledge about an SoS can be stored and manipulated as a knowledge graph, this formulation cannot capture the mathematical nature of the physical interactions. In wicked problems, one finds each sub-domain governed by different mathematical principles. For example, in the electromagnetic domain of radio communication, the system is governed by Maxwell's equations and solving equations for phases and amplitudes, but these radio communications are carrying messages on a social network, which is governed by stochastic processes over a discrete communication graph. In order to integrate these vast differences in fundamental dynamics, we must store information in the unified theory of mathematics provided by ACT. ACT has been developed in response to a call from the mathematical system theorist Jan Willems for a behavioral approach to open and interconnected systems. Toward the end of a long career in mathematical systems theory, Willems proposed an approach to studying complex systems based on the operations of tearing, zooming, and linking. Tearing requires modelers to find joints between the subsystems of a system with concise boundaries between subsystems at these joints. Zooming is the process of focusing on a subsystem and considering its behaviors given the behavior of the environment, where the environment is everything on the outside of the boundary. Finally, linking is the process of considering the effects on those systems in coordinated behavior achieved by sharing state at their boundaries. A goal of ACT is to formalize this approach to system design with mathematical objects including monoidal

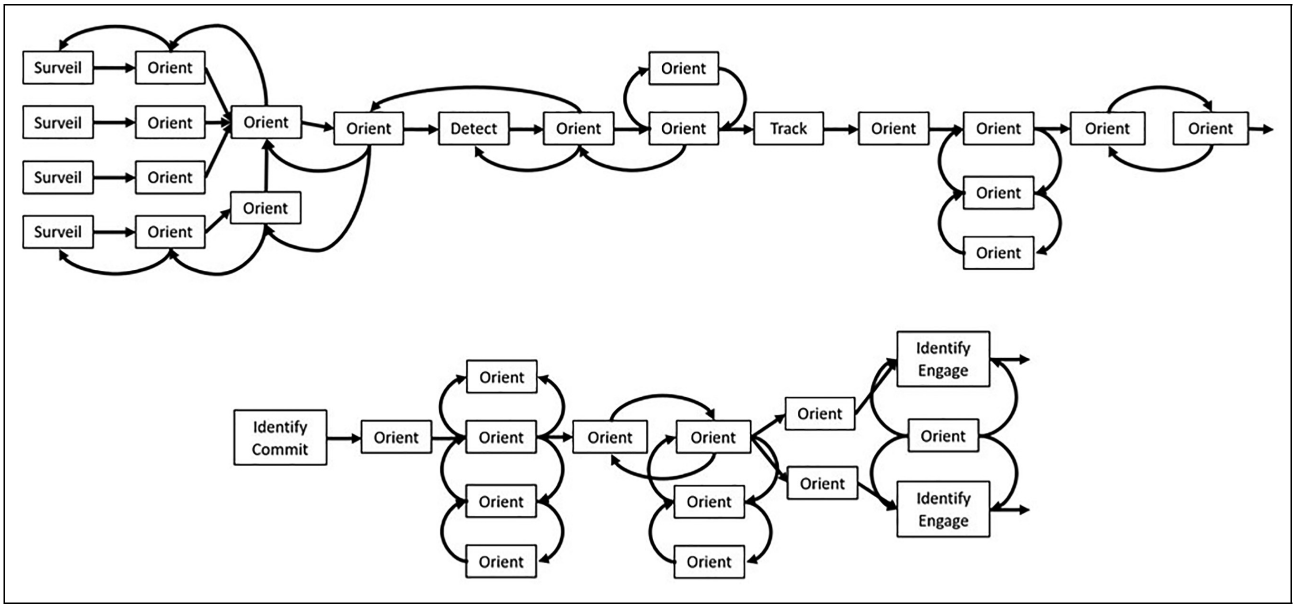

Figure 5. A plausible event chain for the Air Warfare example with complexity (37, 60, ∞) given the looping in sensor network configurations.

categories and operads. An excellent introduction to the field of ACT including the approach to compositional modeling and design is *An Invitation to Applied Category Theory: Seven Sketches in Compositionality*.[17]

Our software approach (SemanticModels.jl is developed on GitHub at https://github.com/jpfairbanks/SemanticModels.jl with documentation hosted at https://aske.gtri.gatech.edu/docs/latest/) implements ACT, which understands physical and computational systems through the lens of *categories*.[18] A category is a mathematical structure, built from *objects* (things) and *morphisms* (relationships between things), where the structure comes from *composition* of morphisms. The traditional presentation of mathematics centers around Set Theory, where the objects are *sets* and the morphisms are *functions* with the traditional definition of function composition. Almost any mathematical object can be viewed as a category, including processes in SE. For example, a *co-design* can be modeled as a category where the objects are resources and the components that provide input resources, produce output resources.[19] ACT seeks universal representations of mathematical knowledge that transcend domains and disciplines.

By taking the ACT perspective, we can build mathematical and computational tools for analyzing systems across diverse domains. The unified framework of categories allows for representing different mathematical frameworks as examples of a common algebraic structure. This unification of heterogenous modeling frameworks allows us to build tools that are specialized enough to exploit structured knowledge about the application area, but general enough to write software against a common interface. One example of where this approach can shine is the modeling of mission threads as a graph. While existing graph-based techniques treat the edges as the primary structure and build hierarchical representations of systems for either understanding or computational efficiency, the ACT approach takes the hierarchical design of the network as primary and deals directly with the consequences of that hierarchy.

Once a system is modeled as a graph, graph analytic techniques, such as pathfinding, centrality, and community detection, can be used to analyze the system. Pathfinding techniques are used to explore paths through a graph. An example of pathfinding is Google Maps where several of the shortest routes between two points are calculated in terms of distance and time. In the mission model, the technique could be used to explore alternate mission threads or event chains. Centrality is used to explore the role of nodes in the mission graph, providing a metric of connectedness. It also entails finding nodes that have significant control or influence, these could be vulnerable choke points in communications or decision-making. Community detection algorithms are based on finding relations and behaviors within the group. These groups could be a structure of resiliency or subsequent failure. The algorithms used for these techniques are mature and available as open source tools, e.g., Apache Spark with GraphX (https://spark.apache.org/, https://spark.apache.org/graphx/). These techniques will provide insight enabling changes to the topology of the mission model and/or mission threads.[20] However, the ACT perspective opens up a whole new set of tools for analyzing systems, such as comparison of networks with metadata via optimal transport.[21]

ME is an inherently multi-domain problem where the dynamics of the problem appear too complex to be mathematically modeled. However, this is true when trying to identify a single set of mathematical rules for modeling all aspects of the mission. When you separate the mission into each domain and model them separately, the ME process is amenable to mathematical analysis. However, since the rules for different domains are diverse, traditional simulation software development techniques fail to give a unified treatment of the system, which is essential for building large-scale software for accurately modeling a complex mission. It is only through the ACT paradigm that we can see how these different mathematical modeling frameworks are examples of categories with various axioms. We can then build software that works with explicit representations of the axioms to build a unified software ecosystem for mathematical modeling and computer simulation of complex multi-domain missions. Building a model is insufficient for ME in wicked problems; the models you build must be used to reason about the world and make decisions. This decision-making process requires that the models be tractable either analytically or numerically. The ACT perspective gives you a framework for analyzing systems with symbolic algebra, with an easy transition to numerical analysis when there is no analytic solution, which is usually the case for wicked problems.

Using an ACT basis for integration enables the rigorous implementation, data management, and extensibility of software. ACT provides for a mathematical foundation used to build the multi-layered mission graph. This foundation can be viewed as graphical maps defining the functional structure and sub-structures of objects and morphisms. The graph then is both the model and the data schema upon which an iterative simulation process can be built. Mission thread construction begins using ACT to create string diagrams[22] facilitating the rigorous and consistent execution of subsequent analyses.

5. Interrogating the mission with modeling, simulation, and analysis looping

Modeling, simulation, and analysis looping (MSAL) is an iterative approach suited to provide a variety of perspectives for quantitatively evaluating the ability of a SoS to meet mission goals.[23] MSAL is created and executed upon a graphical mission model and an initial set of event chains (e.g., Figures 2 and 5) and mission threads (paths) that traverse the model. It is an integration process of independent pieces. As discussed earlier, traditional system modeling languages are not appropriate for an SoS evaluating a complex mission. The mission model should instead be built using a rigorous mathematical specification to guide the integration. The mission model is a schema of a graph database where each node and edge in the model are the basis of independent data stores containing metadata, structured and unstructured data, and essentially extensible sub-databases. Multiple associations, the edges between nodes in the databases, can be made arbitrarily within a mathematical specification. Graph databases have demonstrated scalability, extensibility, and a robust open source toolset that includes visualization, query languages, and graph analytics. They are also readily integrated with machine learning and artificial intelligence tools. The other advantage of a graph database is that a simulation can be built upon the mission model, i.e., the database is a flexible, reconfigurable simulation architecture and framework providing a consistent interface standard. The mission model consists of the following information:

1. Environment data,
2. Relevant systems and resources needed to execute the mission,
3. Context including constraining policy and doctrine,
4. Mission goals,
5. Compilable behaviors, node or edge, to support modeling and/or simulation, which could include conditional probability tables and supporting data, and
6. Live, virtual and constructive data, real-time or historical.

Once the initial event chains and mission threads have been created, MSAL can interrogate event chains by Bayesian Inference when experimental data are available and robust simulations have yet to be developed. The first step of this process is to establish the event chains as paths through the mission graph consistent with mission goals. The second step is to perform functional decomposition and merge the event chain graphs establishing mission processes. The OODA loop provides a means for such a functional decomposition. The event process can then be converted from a cyclic multidigraph into a unique, directed acyclic graph to enable the use of Bayesian techniques. This process will likely create several directed acyclic graphs, i.e., Bayesian networks, for each mission thread. The third step is to collect and analyze existing data sets and establish statistical distributions or conditional probability tables about each mission process. From these analyses, the probability of event success, $P^{Success}_{A|B,C,\ldots}$, may be inferred using Bayesian techniques. If the probability of event success is unacceptably low or there is insufficient data, then experimentation will be used to change mission thread topology. Zakaria[24] discusses a MSAL type approach and proposes a graph-based causal model referred to as action networks. The approach is a scoping technique using ''fuzzy'' data to populate action tables, a construct like conditional probability tables. An action network is a complimentary method in a graph-based toolbox containing structural equation modeling, Bayesian networks, and agent-based simulations.

Experimentation about the event chains can be performed using uncertainty quantification (UQ) coupled with simulation run on scenarios. Scenarios are instances of a mission thread. The use of simulation explicitly introduces a temporal component to the scenario. Agent-based simulations are a key tool to evaluate mission success when inter-entity relationships have similar or greater importance than the performance of individual entities. Agent-based techniques readily deal with networks and inter-agent interactions to include human social factors, non-linearity in agent behavior and/or coupling, or the absence of explicit mathematical solutions. Furthermore, due to their fast-running nature, hundreds to thousands of runs can be realized per day.25[25] The Dempster-Shafer-based UQ engine will propagate both epistemic and aleatoric uncertainties through iterative simulation producing bounds about a multi-dimensional, optimized performance surface.[26, 27] The intent is not to simulate the totality of a mission, but to provide appropriate levels of abstraction and insight where needed within the mission thread due to a lack of empirical data, or a change in context. This simulation-based evidence could lead to changes in mission thread topology, and/or inter-system relationships/behaviors. The optimal solution is not desired; instead, the goal is to minimize uncertainty while maintaining acceptable performance. The characteristics of the uncertainty bounds about the optimized solution are more important to decision makers than the absolute nature of an optimized performance prediction. The use of UQ in simulation is mature technology; an example is DAKOTA from Sandia National Laboratories (https://dakota.sandia.gov/, page 53 of the theory manual).

Various components of MSAL have been demonstrated; Mabrok and Ryan[28] used ACT as the mathematical specification to create a robust foundation for model-based SE focusing on the relationships between components explicitly including requirements. ACT has also been used to rigorously couple agent-based simulation with Bayesian inference to create hybrid methods consistent with MSAL.[29] Henkel et al.[30] applied graph database techniques to enable robust queries and data analyses in Biology. Key to the approach was maintaining relationships between the models, simulations, and metadata to facilitate reuse. Using a MSAL type scheme, Saleem et al.[31] built a graph-based framework to model an urban water supply that addresses many ME limitations previously stated. The graphical model has nested supply and demand entities consisting of a macro-layer, the water supply and neighborhood, a meso layer of the water works, buildings and housing, and a micro-layer of occupants, and water resources. The number of entities in their model is 63 on the demand side and 12 on the resource side, where the edges of the graph are the flow of water from the supply side to the demand side. Entity and edge characteristics are quantified with equations, and a simulation built upon the model architecture. Scenarios represent the paths of water flow with defined context. UQ is conducted bounding the simulation results. Validation is conducted using comparison to existing real-world data sets. This emerging work gives confidence to build the multi-faceted MSAL.

6. A graphical example

In this section, the mission thread, ACT, and MSAL concepts are illustrated in an example using a city and students. The city is based on the *city anatomy framework*,[32] where the city anatomy provides ‘‘a hierarchically sound and well-established description, identification, nomenclature, and classification of all city systems, subsystems and interactions....’’ The framework is language-based and, at the highest level, composed of three interacting layers: the city *structure*, *interactions*, and *society* as shown in Figure 6.

A graphical instance of the framework is shown in Figure 7, containing 68 nodes and 211 edges. This graph is math-based and testable. The edges of the graph represent flows (e.g., electricity and water), interactions, or hierarchical relationships (i.e., ‘‘subsets of’’). Specific details of the city like roads and buildings would be sub-structures embedded within the graph. This graphical instance is the basis for a conceptual model and subsequent abstraction.

A notional mission involving college students on the Georgia Institute of Technology (GaTech) campus will be used as an example for creating event chains; the first step in implementing MSAL. The example will be static and lead to the creation of an OODA-based event chain. The mission graph will be multi-layered and consistent with the city anatomy framework.

6.1. Mission model

The mission model for this example is a subset of Figure 7. The mission environment is Atlanta Georgia, specifically the GaTech neighborhood. Using the city anatomy taxonomy, this example explicitly represents:

- Structure layer components: *Built Domain*—neighborhood, buildings, and dwellings; *infrastructure*—*mobility network* comprised of streets, bike paths and walkways, and *communication network.*
- Society layer components: Citizens—students and professors at GaTech.
- Interaction layer components: Functions—working, education, and transport.

Atlanta is divided into 12 Districts and has 242 neighborhoods. The GaTech (http://documents.atlantaregional.com/

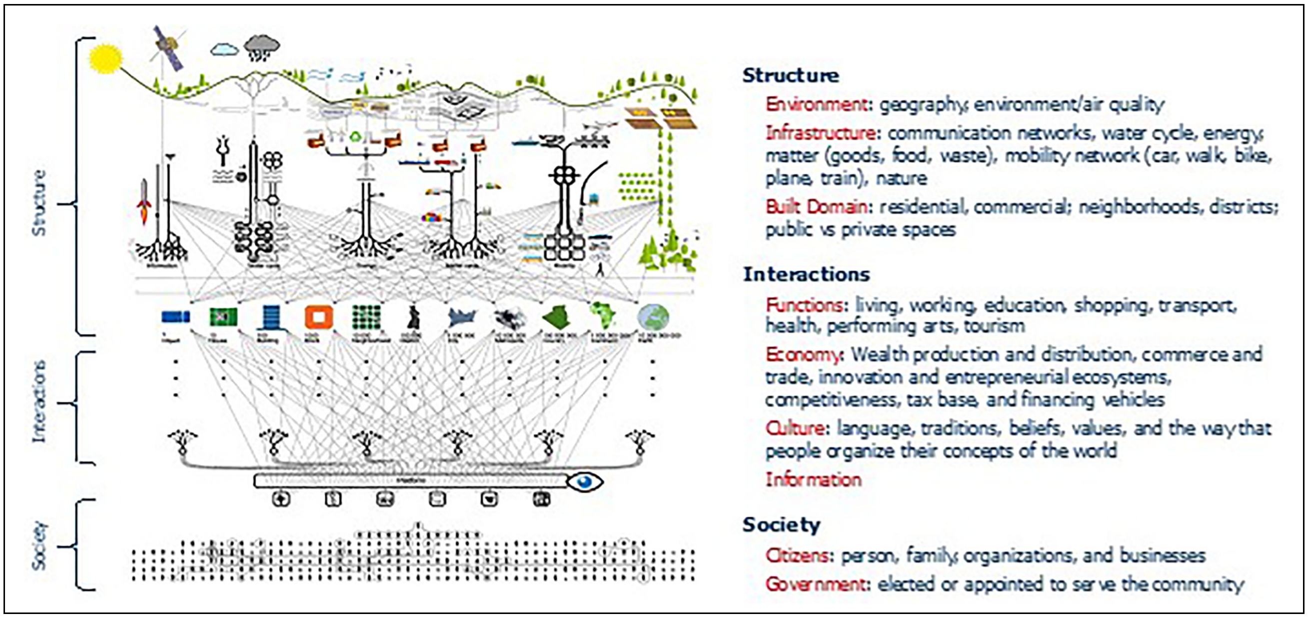

Figure 6. The city anatomy framework, as basis for modeling a city.

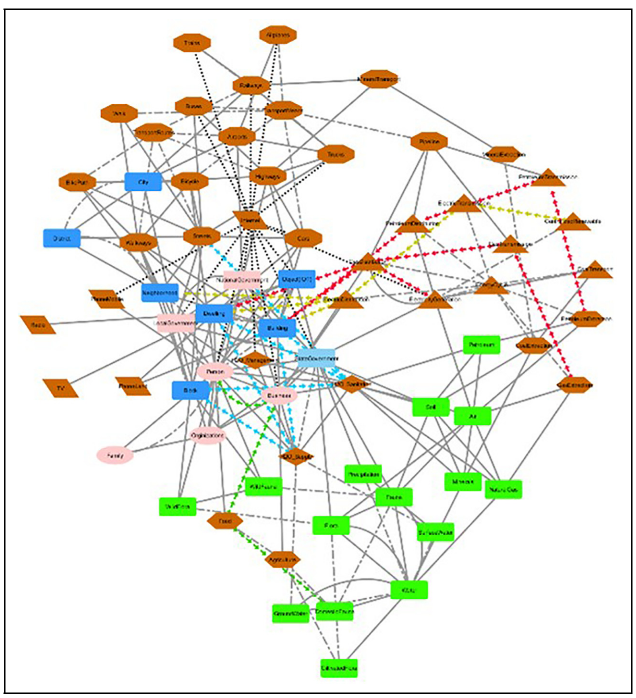

Figure 7. A graphical instance of the city anatomy framework containing all the information in Figure 6.

NN/Profiles/AtlantaProfiles/E02.pdf) neighborhood population is about 8000 with approximately 90% being college aged. There are around 970 dwellings where nearly 90% are non-family households. Figure 8 is a map of the GaTech campus from Open Street map (https://www.openstreetmap.org/?edit_help = 1#map = 16/33.7757/-84.3999), along with a close-up view of the center of campus. This map is composed of 45 nodes and 592 edges along with extensive metadata for each. This is significant complexity, and parsing the data is necessary for subsequent analyses.

6.2. *Mission threads*

The mission is defined as groups of students from a class working to solve a homework assignment. One of the groups decides to cheat by plagiarizing from classmates, with their mission goal being to fool the professor that their work is original. ACT will be used to rigorously define the mission schema, a precursor to the mission thread. Figure 9 shows three representations of the mission schema:[33] 9a is programming syntax that could be used by a subject matter expert; 9b is an algebraic expression that would be easy to manipulate with algorithms; and 9c is the string diagram that is easy to visualize and understand. Each string is an entity type, and each box is a process that matches entity production with requirements. This string diagram also necessarily defines a database schema and can be used to construct a mission simulator. The ACT approach to simulation design is implemented for continuous time dynamical systems in the software package AlgebraicDynamics.jl. A detailed description of converting a string diagram-based system representation into a computational model is provided in Libkind et al.[34] The AlgebraicJulia software supports discrete time dynamical systems approaches for representing agent-based models. The generating objects are taken as the states of the agents and the generating morphisms are the state transitions that can occur in the system. From this specification of the

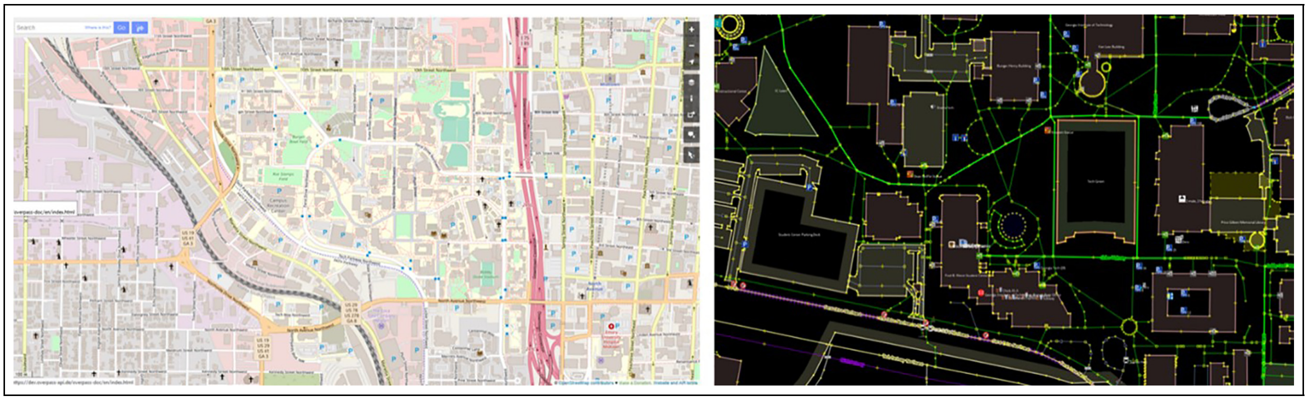

Figure 8. Images of the GaTech campus from Open Street Map. The image on the right shows the detail in the data set consisting of the node and edges representing buildings, streets, walkways, etc.

(a)

```
d = @program Transit
      (s₁::Team, s₂::Team, p::Prof)
begin
  s₁, a₁ = class(s₁, p)
  s₂, a₂ = class(s₂, p)
  s₁ = north(s₁)
  sol₁ = solve(s₁, a₁)
  s₁ = east(s₁)
  s₂ = east(north(s₂))
  sol₂ = solve(s₂, a₁)
  s₁ = west(s₁)
  g₁ = gradea(sol₁, p)
  g₂ = gradea(sol₂, p)
  return g₁, g₂
end
```

(b) $(\mathrm{id}_{\mathrm{Team}} \otimes \mathrm{id}_{\mathrm{Team}} \otimes (\Delta_{\mathrm{Prof}} \cdot ((\Delta_{\mathrm{Prof}} \cdot (\Delta_{\mathrm{Prof}} \otimes \mathrm{id}_{\mathrm{Prof}})) \otimes \mathrm{id}_{\mathrm{Prof}}))) \cdot (\mathrm{id}_{\mathrm{Team}} \otimes \sigma_{\mathrm{Team,Prof}} \otimes \mathrm{id}_{\mathrm{Prof} \otimes \mathrm{Prof} \otimes \mathrm{Prof}}) \cdot (\mathrm{class} \otimes \mathrm{class} \otimes \mathrm{id}_{\mathrm{Prof}} \otimes \mathrm{id}_{\mathrm{Prof}}) \cdot ((\mathrm{north} \cdot \Delta_{\mathrm{Team}}) \otimes \Delta_{\mathrm{Assgn}} \otimes (\mathrm{north} \cdot \mathrm{east}) \otimes \Diamond_{\mathrm{Assgn}} \otimes \mathrm{id}_{\mathrm{Prof}} \otimes \mathrm{id}_{\mathrm{Prof}}) \cdot (((\mathrm{id}_{\mathrm{Team}} \otimes (\mathrm{east} \cdot \mathrm{west} \cdot \Diamond_{\mathrm{Team}}) \otimes \mathrm{id}_{\mathrm{Assgn}} \otimes \mathrm{id}_{\mathrm{Assgn}} \otimes \mathrm{id}_{\mathrm{Team}}) \cdot (\mathrm{id}_{\mathrm{Team} \otimes \mathrm{Assgn}} \otimes \sigma_{\mathrm{Assgn,Team}}) \cdot (\mathrm{solve} \otimes \mathrm{solve})) \otimes (\mathrm{id}_{\mathrm{Assgn}} \otimes \sigma_{\mathrm{Assgn,Prof}} \otimes \mathrm{id}_{\mathrm{Prof}}) \cdot (\mathrm{gradea} \otimes \mathrm{gradea})$

(c)

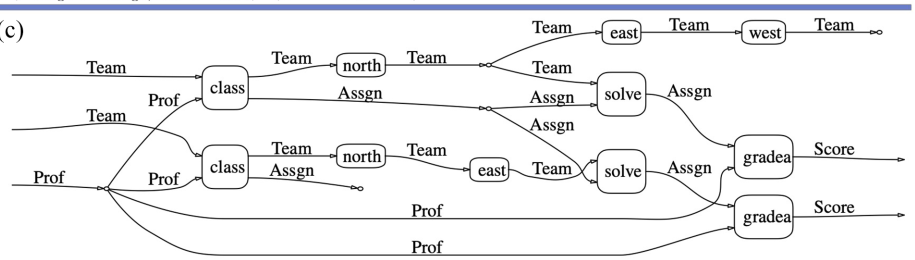

Figure 9. An example mission thread of students plagiarizing an assignment in a college class. The mission thread is shown in three representations of a schema-level mission thread: (a) a programming syntax that is easy to write, (b) a formula that is easy to computationally manipulate, and (c) a string diagram that is easy to visually interpret and understand.

states and transitions that are possible, a discrete time dynamical system can be generated programmatically. These systems are then simulated using a time-stepping algorithm.

The next step is to explicitly define mission resources and relations. The resources (top of Figure 10) are the students, *Team*, taking a class together; the professor, *Prof*, teaching them; and the homework assignment, *Assgn*. The product of the mission is the grade the professor gives the assignment, *Score*. There are two subsets of *Team*, S_1 and S_2. The relations are below the resources, including *class*, *north*, *east*, *solve*, *west*, *gradea*, and *plagiarize*. *Class* is the process where the professor and students create a learning environment resulting in an assignment given to *Team*. *North*, *east*, and *west* are the motions the *Team* takes transiting about the campus. *Solve* is the process of completing the homework. *Gradea* is the professor's process for grading the homework. *Plagiarize* is the process of team S_2 copying the homework from an element of S_1. The mission will track the movement of the individuals of Team and Prof through the campus as the students complete their assignment and receive a grade.

The *Team* and *Prof* transit on *streets*, *walkways*, and *bike paths* between their *dwellings* and *buildings* containing classrooms, offices, and study areas. Once *Team* and *Prof* arrive at the appropriate building they participate in the *work* and *education* interactions. Transit is represented by a series of event chains composed of the primitive movement operations *north*, *east*, *and west*. Each of the processes of *class*, *solve, gradea*, and *plagiarize* is represented by an event chain. A nearest neighbor analysis of the city anatomy graph show that *persons*, *buildings* and *dwellings* may have significant impact on the outcome of the event chain. The integration of these event chains would then become the mission thread. Based on the number of possible primitive events, there is a combinatorial explosion of mission complexity.

6.3. Mission threads and OODA

To complete a mission thread, a functional decomposition of the schema must be conducted. For this example, we will build an event chain, a subset of the overall mission thread. An OODA-based event chain is a directed acyclic

```
@present Transit(FreeCartesianCategory) begin
  # Generating Objects
  (Person, Team, Material, Homework, Prof, Score, Test, Assgn)::Ob
  # Generating Morphisms
   north::        (Team → Team)
   south::        (Team → Team)
   east::         (Team → Team)
   west::         (Team → Team)
   meet::         (Team⊗Team → Team⊗Team)
   exchange::     (Team⊗Team → Team⊗Team)
   acquire::      (Team⊗Material → Team)
   comm::         (Team⊗Team → Team⊗Team)
   class::        (Team⊗Prof → Team⊗Assgn)
   exam::         (Team⊗Prof → Team⊗Test)
   presentation:: (Team⊗Prof → Team⊗Assgn)
   gradea::       (Assgn⊗Prof → Score)
   gradet::       (Test⊗Prof → Score)
   gradeh::       (Homework⊗Prof → Score)
   solve::        (Team⊗Assgn → Assgn)
   take::         (Team⊗Test → Test)
   obsfucate::    (Assgn → Assgn)
   plagiarize::   (Team⊗Assgn → Assgn)
end

# Selected Generating Axioms
north · south == id
south · north == id
east  · west  == id
west  · east  == id
(copy⊗id)·copy   == (id⊗copy)·copy
(merge⊗id)·merge == (id⊗merge)·merge
```

Figure 10. An example category presentation. One lists the concepts and relationships between those concepts along with any axioms of the theory. These axioms allow symbolic computation techniques to reason about models analytically before running an explicit simulation of the model numerically.

graph where the edges represent the movement of people or information consistent with the schema, shown in Figure 9. There are a set of edges representing *Team* transiting to and from the processes *class, solve*, and *plagiarize*. There are a set of edges representing *Prof* transiting to and from *class* and *gradea*. There are also two types of edges: external communication between sets S_1, S_2, and/or *Prof*, and internal communication within a set. The nodes for S_1, S_2, and *Prof* are Observe, Orient, Decide, and Act where the Orient nodes are as follows:

1. Parse the physical and natural environment relevant to the moment,
2. Parse the human environment relevant to the moment,
3. Parse and analyze the Observe data,
4. Mine historical data,
5. Reassess the ability to meet the mission goal and effects chain goal(s).

The Orient nodes are the essence of learning from observation and experimentation.

The concept of creating an OODA-based event chain will be demonstrated on the process *class*. Using *Prof* as an example, a plausible functional decomposition for *class* could begin as

- *Prof* walking to *Class*: Act within the *transport* function,
- *Prof* checks the IT infrastructure and reassess the ability to meet the lesson goal: Orient, which transitions from *transport* to *working*,
- *Prof* observes *Team*: Observe,
- *Prof* processing state of *Class* and *Team*: Orient 1, 2, and 3,
- *Prof* decides to start lesson: Decide,
- *Prof* teaching in *Class* with *Team*: Act,
- *Prof* teaching in *Class* with Team: Orient 2, 3, and 5.

The Orient function could be represented as multiple entities or bundled as a single entity type. Figure 11 is an instance of an event chain for *Class*. The class begins as above with the entrance of the entities of *Prof* and *Team*. In class, a learning environment is established, teaching material presented, a question-and-answer session followed with an assignment given to *Team*. Team breaks up during class into subsets, S_1 and S_2. The class ends with a communication between students in S_1 and S_2 requesting assistance to cheat; then, S_1 and S_2 transit away in two different directions.

OODA is about creating processes and implementing procedures that cycle through the loop quickly and ultimately lead us to action, whether it is automated or not. Consider the plagiarism example of Figure 9, once *Prof*

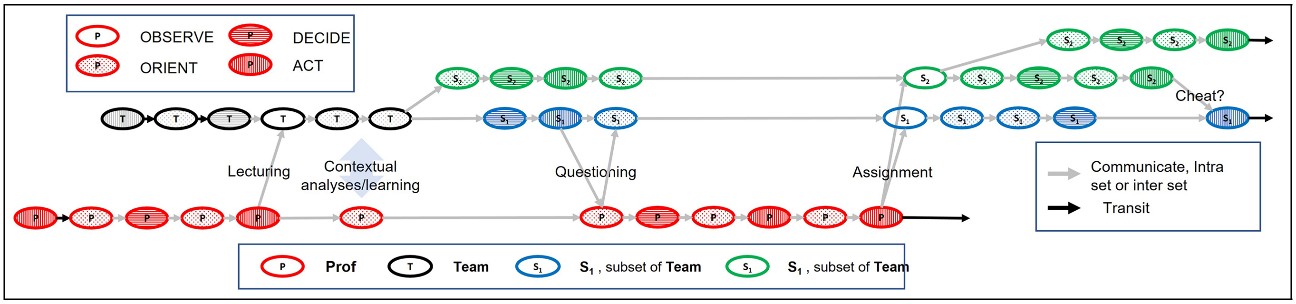

Figure 11. An event chain for *Class*.

has observed the *Team*, it is parsing the physical and natural environment, parsing the human environment, and parsing and analyzing the Observe data that lead to the conclusion that the lesson, Act, can begin. It is this Orient processing that enables the action to occur. In a similar manner, *Team* observes the lecture and the subsequent Orient processes the lesson, i.e., learning, leading to the action of questioning. Likewise, when a member of *Team* S_2 cheats, they decide after reassessing their ability to successfully complete the homework assignment. The benefit is from Orient processes, which provide the insights that inform the decisions and subsequent actions needed. Where Observe, Decide, and Act are system functions, Orient is multidimensional.

6.4. Mission mathematics

In the ACT paradigm, you define a model as a syntactic expression that describes a model as a formula, then give that model semantics that you can view as a meaning, behavior or implementation. In terms of ME, the wiring diagram (or equivalent program or formula) is the syntactic representation of the mission, and the semantics of the mission are given by a relational algebra database instance or a run of a simulation in a traditional M&S context.

For example, a mission thread containing movement operations is a relation on places that satisfy the constraints of that thread. If the mission thread was $p = north \cdot east \cdot north$, then the semantics of that mission thread is the set of all pairs of points (x_1, y_1), (x_2, y_2) such that there is a path $(x,\ y+1) \rightarrow (x+1,\ y+1) \rightarrow (x+1,\ y+2)$ in the transportation network. In a city aligned to a grid, we have the axiom that $north \cdot east = east \cdot north$ so an automated reasoning algorithm could compute analytically that $north \cdot east \cdot north = north \cdot north \cdot east$ and thus would know that the semantics of a path p is given by the set $\{(x,\ y),\ (x+1,\ y+2) \mid x, y\}$. The ACT paradigm connects syntactic descriptions of models to their mathematical behavior by the principle of compositionality. Let F denote the map from descriptions of systems to their behaviors, we say that F is compositional if $F(f \cdot g) = F(f) \cdot F(g)$ for all pairs of models f, g. This property can be characterized by F preserving the algebraic structure of composition. Algebraic properties and maps that preserve them are essential to the mathematical analysis of systems.

By formalizing our ME frameworks as algebraic objects (categories), we can build computer algebra systems that can analyze mission threads within those frameworks. When implementing the semantics of ME in a software system, a relation database can store the data and database joins implementing the composition rule for the semantics. In the example above, the composition of path is implemented as a database join because if you want to compose a path $p \rightarrow q$ with a path $r \rightarrow s$ you are looking for all the paths $p \rightarrow q$, $r \rightarrow s$ where $q = r$. This is precisely the notion of a database join and could be implemented in SQL as **`select t1.start, t2.end from paths as t1 join paths as t2 on t1.end == t2.start`**. In this way, the map F transforms descriptions of paths into relations on pairs of points that turns composition of paths into joins of relations. When we say that ME software can generate a simulation of the mission thread, it is exploiting these structure-preserving maps to go from wiring diagrams of the mission thread to a computer program that computes the semantics of the mission thread. In the example above, that means converting sequences of directions into SQL queries that compute sets of paths.

When implementing the semantics of a mission thread in real-world data, you have to mine the structures from data you have available. For that purpose, we turn to the Open Street Map as a source for transit networks. The mission thread specifies that a set of event chains for *Transit* would be a set of paths along *streets*, *walkways*, and *bike paths* through the campus. There would be the set of paths for *Prof* and a set of paths for each element (student) in the set *Team*. These paths are obtained from shortest path applications, and a temporal component could be added with routing applications, e.g., the proprietary Google Maps or the open source Grasshopper Maps. The parsing and interrogation of Open Street Map data can be readily achieved by the use of open source tools like OSMnx

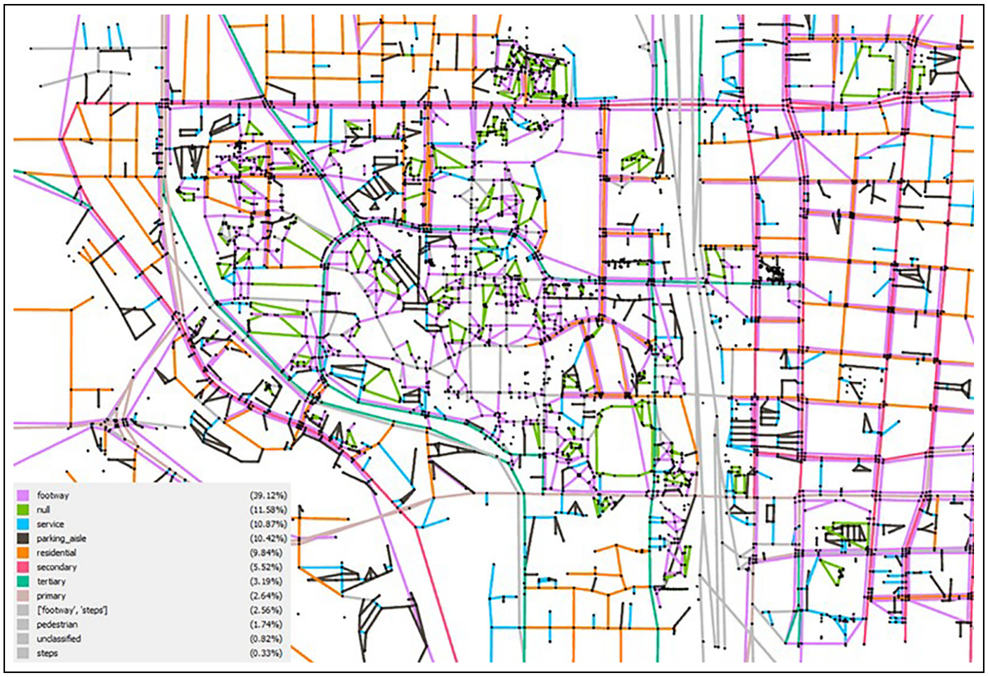

Figure 12. The mobility network in the GaTech neighborhood, parsed from Open Street map, is presented as an undirected graph with 5057 nodes and 6373 edges.

tool.[35] A parsed data set from OSMnxset from which routing analyses could be conducted is shown in Figure 12. In this data set, there are 5057 nodes and 6373 edges and 12 defined categories of streets, walkways, bike paths, parking lots, etc. For brevity, we will not execute the routing analysis in this paper.

7. Trust, an overarching measure of mission confidence

DoD ME requires successful integration and interoperability of multiple complex systems, which were not necessarily designed to communicate with one another. There has been little research with respect to the integration and testing of SoS and, in particular, verification and validation activities to assure that the SoS successfully achieves the mission and satisfies requirements.[36] As discussed in Section 2, SoS V&V is challenging. In Luna et al.,[37] it was recognized that traditional SE methodologies for integration, verification, validation, test, and evaluation are understood because they are focused on expected outcomes (behavior, capabilities). However, these methodologies may not apply to SoS since emergent behavior and knowledge are needed in test scenarios. Similarly, it is accepted that V&V for systems is ''generally possible''; however, for SoS, it is more challenging due to complexity, unspecified behaviors and requirements, and the potential for unintended consequences.[38]

In the absence of approaches to enable virtual testing for SoS, we are left with brute-force methods which are severely limited in the range of conditions they can test.[39] Zeigler goes on to claim that a key root cause of limitations in current V&V approaches to SoS is that they are not based on a general M&S framework capable of expressing the interaction of decision logic, discrete events, and continuous dynamics systems. While Zeigler proposes the DEVS formalism for architecting the simulation software, others propose new SoS testing processes to reduce integration risk.[36,40]

Traditional forms of V&V are problematic for SoS, notably the large number of potential system states, network interactions leading to emergent behavior, flexible cohorts of constituent systems, dynamically evolving requirements, rapidly changing technology, and non-traditional acquisition models.[40] As discussed in *Assessing the Reliability of Complex Models*,[41] given the inevitable flaws and uncertainties in models, it is more appropriate to think about how computational results should be viewed by those who wish to act on them, rather than thinking about the results as a binary answer. The report focused on physics-based and engineering models; however, much of their key principles and practices apply more broadly, including SoS. For example, V&V are not yes/no questions with yes/no answers, rather they are quantitative assessments of differences. Best practices identified in National Research Council[41] suggest using a broad range of physical observation sources so that the accuracy of a model can be checked under different conditions and at multiple levels of integration.

To address the challenges with SoS V&V and embrace the recommendations of the National Academies report, we propose trust as an alternate concept to provide confidence to decision-makers.

7.1. Defining trust

Highly complex missions are inherently difficult because of incomplete, contradictory, and changing requirements. Furthermore, social complexity mean solutions to ME problems should not be evaluated as ''true or false,'' but rather ''better or worse.'' Thus, trust as a concept for ME should not be a binary concept; rather, it should capture the uncertainty of a multi-faceted problem.

We can think of trust as multidimensional including characteristics such as dependability, security, reliability, quality, and availability within a specified context.[42–44] The idea that trust is a level of confidence is captured by Voas et al.:[45] the probability that the intended behavior and the actual behavior are equivalent given a fixed context, fixed environment, and fixed point in time. Recognizing trust is multidimensional; NIST defines it as ''... the demonstrable likelihood that the system performs according to designed behavior under any set of conditions as evidenced by characteristics including, ... security, privacy, reliability, safety and resilience.''[46] We can also think about trust in levels, for example in human society is might include security, institutional, and social[47] or for systems it might include technical, computational, and behavioral.[48] Finally, Terzis[49] defined it as security-oriented and non-security-oriented. In security-oriented, trustworthiness is an absolute property that an entity either has or does not have, which can be determined using the credentials an entity possesses. Non-security-oriented trust adopts a social science view, where trust is determined on the basis of evidence (personal experiences, observations, recommendations, and reputation) and is situational, meaning an entity's trustworthiness differs depending on the context of the interaction.

7.2. Layers of trust

By combining the idea of dimensions and levels, we can define a set of trust layers for ME. We will adopt the layered trust framework defined by Yan et al.,[50] where the layers work together to create an environment in which things and humans can interact and make trustworthy decisions. The layers include the following:

- i Perception layer perceives the physical environment including human social life,
- ii Network layer transforms and processes the perceived environment data,
- iii Application layer offers context-aware intelligent services in a pervasive manner, and
- iv Trust management represents the cyber-physical social relationships that connect layers. ''Ensuring the trustworthiness of one ... layer (e.g., network layer) does not imply that the trust of the whole system can be achieved.''50

In a recent SoSE information exchange, an instantiation of the digital ecosystem for ME was defined as having three activities:[11] (1) transparency and sharing of data and models; (2) collaborative environment to install simulation and software tools for ME analyses; and (3) curation of trustworthy data—accuracy of analyses depends on pedigree of data. The goal of this digital ecosystem is to provide decision-makers with confidence. While only the last bullet specifically mentions trust, these activities together create a trustworthy ME ecosystem. As indicated below, we can map each activity to one of the layers in our framework.

- i Perception layer: curation of trustworthy data—Where to Trust
- ii Network layer: collaborative environment—When to Trust
- iii Application layer: transparency and sharing of data and models—What to Trust

7.3. Trust example: simulation ensembles

Our approach to defining layers of trust to gain better insight into model behavior does not advocate for a specific simulation technique. Rather, it is a framework to bring together a broad range of simulations and understand their strengths and weaknesses. Simulation ensembles bring different levels of abstraction/resolution and the MSAL looping framework builds insight. An ensemble is a toolbox of different simulators based on a common mission model and iteratively run on or about a common mission thread. The ensemble is used to interrogate a mission from different perspectives and levels of abstraction. When ensembles are run in conjunction with UQ, mission performance surfaces can be identified for more detailed investigation.

Ensembles have been used to address the complexity of climate and to interrogate uncertainties at various length (global versus regional versus local) or time scales (a single hurricane versus decades).[51,52] In the following sections, we use the example of hurricane forecasting to describe the layers of Trust, and show how they can be used to provide stakeholders confidence in the simulation results and manage risk.

7.3.1. Trust at the application layer—"what to trust." This is where the individual simulations reside (e.g., agent-based,

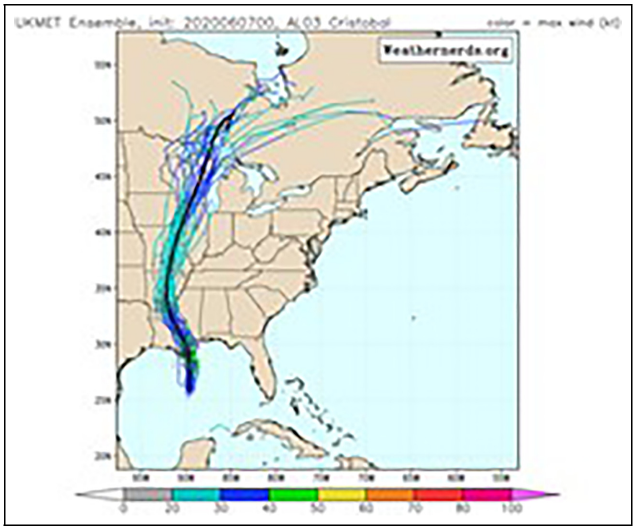

Figure 13. Parameter sensitivity analysis from a single simulation.

equation-based, and AI-based) that execute the mission model. Since no one simulation contains everything important to the mission model, it is conceivable that strength of trust comes from having many simulations with different assumptions and world context. In other words, an ensemble of simulations with different perspectives has the effect of reducing risk by giving greater understanding to sensitivity and uncertainty. First, each simulation is tested with the same data set (parameter distributions, not scenarios). Single- and multi-parameter sensitivity studies are used to assess the most impactful parameters for subsequent study. Optimization provides insight into areas of high gradients and parameter interactions within the performance space. UQ defines confidence bounds about the performance space for each simulation.

For example, Figure 13 shows the effects of a parameter study with a single simulator. Each prediction of the path of hurricane Cristobal, from UKMET (A medium-range (3 to 7 days) numerical weather prediction model operated by the U.K. METeorological Agency. It has a resolution of 75 kilometers and covers the entire northern hemisphere. Forecasters use this model along with the ECMWF and GFS in making their extended forecasts (3 to 7 days). https://forecast.-weather.gov/glossary.php?word = ukmet) simulator is shown as a separate line; these are instances of bounding mission threads. Since the same input distributions should yield similar results, all of the ensemble simulations should show similar trends (e.g., uncertainty volumes). If one of the simulations produces results very different that the rest of the ensemble, there is reason to question that simulation. In other words, the comparison of ensemble results leads to an understanding of which simulations to trust, and how much.

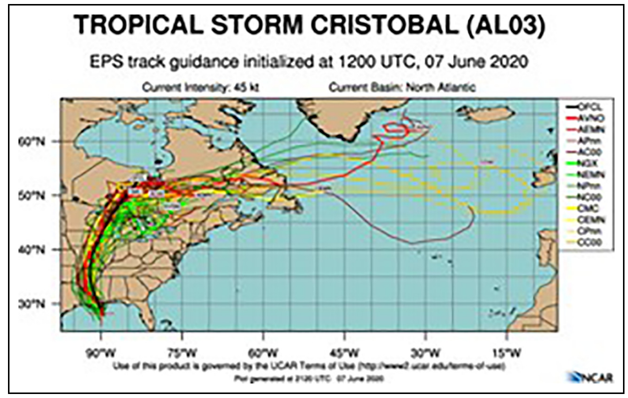

Figure 14. Ensemble hurricane prediction.

7.3.2. Trust at the network layer—"when to trust.". This is where the mission graphs and data stores reside. The trust network contains the input and output data from all MSAL and ensemble runs. A mission graph is composed of systems, the environment, policy and doctrine, and connecting relationships with significant metadata descriptors. The trust network is the domain for mission analytics, e.g., testing the mission graph, post-processing ensemble runs, and creating new mission threads. For the city anatomy example in Section 6, this would include the structure, society, and interactions dimensions and capture both a visual and mathematical behavior of people and mobility. At this layer, mission graphs are structured using ACT, and transformed using graph analytics (quantifying/assess topologies, centrality, clustering). Temporal changes to mission topology, based on context, are one particular focus of trust in this layer. While individual models and simulations could be proprietary, the data in the trust network should/could be owned/controlled by the mission stakeholder(s) and available for all with a ''need to know.'' This would provide for continuous mission analysis and improvement in mission capabilities.

7.3.3. Trust at the physical perception layer—"where to trust.". This is where the mission threads reside. A mission thread is a sequence of actions/processes with quantifiable outcomes. There are many plausible mission threads that can be executed, and each thread has a temporal component. Simulator-specific scenarios are created for each mission thread along with establishing the input parameters. An ensemble of simulators is run on each mission thread, and the aggregate outcome is used to establish error bounds. It is by considering the ensemble of all possible mission threads that provides the best insight, and enables us to derive a greater understanding of Trust.

To illustrate this concept, consider how simulation is used in hurricane tracking. In Figure 14, an ensemble of simulations is run on the mission threads, and the aggregate outcome shows the error bound of all possible outcomes. The approach would be consistent with the ensemble-based probability swath Ortt[51] used to express uncertainty in hurricane tracks. The ensemble is from the European Center for Medium-Range Weather Forecasts (ECMWF), which has pioneered a system to predict forecast confidence. The Ensemble Prediction System (EPS) shows us that over time context evolves and dispersion increases.

7.3.4. Trust management layer—"how to trust.". This layer can be thought of in terms infrastructure and the systems which check that programs, e.g., databases, queries, and analytics are sound. Using ACT in constructing models results in more precise constraints about what kinds of mission threads are valid scenarios for your simulation. A software system that uses the principles of ACT can validate these scenarios before executing them based on a logical notion of well-formed expressions. By designing a category whose objects are types of values and whose morphisms are processes, where each morphism respects the domain knowledge of experts in the systems under examination, computational systems can automatically validate statements about these systems including existence of particular mission threads. This computational validation yields higher trust in the resulting M&S systems.

ME software can check the well-formedness of mission threads by checking that whenever two processes are composed sequentially, the output type of the first process matches the input type of the second process. The structure of an ACT-based theory contains all the information necessary to make these type judgments. That a mission thread is well-formed is not sufficient to say that it is relevant to a ME analysis or that a set of well-formed mission threads is exhaustive, but it is necessary for a mission thread to "make sense" within the scope of the system. This requirement can allow mission thread planners to avoid the consideration of meaningless scenarios in a machine enforceable way. Many simulation developers in DoD communities will object to analyses conducted with a simulation software because the scenarios analyzed "don't make sense from a domain perspective." These objections are informal arguments against certain mission threads, and by formalizing this process as checking the well-formedness of expressions, we can automate those judgments, which accelerate ME timelines. While these constraints are sometimes frustrating to developers, they are very valuable for ensuring the correctness and integrity of large-scale software systems. While traditionally these type checking techniques are only used in the design and implementation of programming languages, the ACT approach widens their applicability. Our ME approach applies these techniques to the analysis of mission threads.

8. Conclusions

This paper presents a robust approach to multi-disciplinary, ME beginning with a military example. The concept is then applied to a Smart City with insights into trust in weather forecasting.

ME is the definition, identification, assemblage, analysis, and quantification of an SoS to achieve measurable desired results and, therefore, mission success. The mission is constrained by dynamic operational context and can be inherently complex. Missions are graph-based where the number of edges can approach the number of entities squared and the maximum number of paths can approach the number of entities factorial. From the combinatorics, many plausible mission threads are possible for a given mission. However, not all paths are feasible or desirable due to the governing rules of the mission domain. Missions tend to be layered and multidimensional. They are layered in terms of abstraction; a node in a mission graph can contain very detailed sub-structure. They are multi-dimensional in that several interrelated functions can occur simultaneously; these functions appear as edges in the mission thread. ACT seeks universal representations of mathematical knowledge that transcend domains and disciplines. The ACT approach is inherently computational and universal, which makes it a candidate framework for studying ME. ACT is suited to rigorously define a mission schema. This mathematical basis of the schema necessarily defines a corresponding database schema, and the architecture of a mission simulator.

Mission success is dominated by the inter-system relationships. The key to quantifying these interrelationships is an OODA-based decomposition of mission threads, and event chains are Boyd's Orient function. Orient is used as an edge in a graph to represent the function of communication. Communication encompasses the communication infrastructure, the syntax and semantics of the message, timeliness and trustworthiness, and verification that the meaning of the message was successfully conveyed. In addition, Orient has several entity (nodal) functions related to situational awareness and information processing, defined as

- Parse the physical and natural environment relevant to the moment,
- Parse the human environment relevant to the moment,
- Parse and analyze the Observe data,
- Mine historical data, and
- Reassess the ability to meet the mission goal and effects chain goal(s).

The ability to parse the physical and natural environment (genetic heritage) and human environment (cultural conditions) relevant to the moment provides necessary context enabling evidence-based trust assessment. In complex missions, where the environment is in constant change, these Orient sub-functions reduce uncertainty, and enable better decisions and informed actions.

The use of MSAL techniques along with simulation ensembles provides a variety of perspectives for quantitatively evaluating the ability of an SoS to meet mission goals. These techniques are ideally suited for mission analyses where initial conditions and parameters are uncertain. Graph theoretic and ACT-based constituent capabilities of MSAL have been demonstrated in the literature enabling integration of a robust MSAL toolbox. While validation of ME analysis is problematic due to limited empirical mission data sets, the use of MSAL with ensembles enables alternative approaches to establish confidence in the M&S results. Hurricane forecasting is an example where the use of ensembles and visualization techniques enables quick assessment of high-probability mission threads (storm Tracks).

Since missions are inherently uncertain due to complexity and the dynamic nature of context, the broad issue of trust across a mission knowledge base and mission level analysis become vital to decision-making. Trust can be measured for both security and non-security properties of missions, yet understood at the mission level. A trust framework is proposed that is comprised of four layers: application, network, physical perception, and trust management. For mission models and simulations, the application layer evaluates single simulators and quantifies uncertainties. The network layer is where the mission graphs and data stores reside. The trust network contains the input and output data from all MSAL and ensemble runs. The physical perception layer uses an ensemble approach of multi-simulators running the same mission threads similar to the technique used for hurricane prediction. Convergence of the results over time builds Trust. The network management layer pulls together the three layers and is constrained by an ACT basis. Along with the network layer, the network management layer provides for transparency and sharing of data and models, a collaborative environment to install simulation and software tools, and curation of trustworthy data. Probabilistic visualizations from ensemble result that enable decision-makers to quickly grasp which mission threads are feasible with what level of risk is an emerging and important area of research. The goal of MSAL is not to find the perfect answer; rather, it provides a quantifiable, testable approach that enables leaders, executives, scientists, engineers, and warfighters to make better-informed decisions.

Acknowledgements

The authors wish to thank our friend and colleague Neil Baron for reviewing the manuscript and offering many useful thoughts for us to ponder.

Funding

This effort was funded by the US DoD, OUSD(Acquisition & Sustainment)/ASD(Acquisition), ME and Integration under contract HQ0642930737.

ORCID iD

Margaret L Loper https://orcid.org/0000-0002-0977-696X

References

1. Gold R. Mission engineering. In: *Proceedings of the 19th annual NDIA systems engineering conference*, 2016, https://ndiastorage.blob.core.usgovcloudapi.net/ndia/2016/systems/18950_RobertGold.pdf
2. Sousa-Poza A. Mission engineering. *Int J Syst Syst Eng* 2015; 6: 161–185.
3. Rittel HW and Webber MM. Dilemmas in a general theory of planning. *Policy Sci* 1973; 4: 155–169.
4. Liu L, Loper M, Ozkaya Y, et al. Machine to machine trust in the IoT era. In: *Proceedings of the 18th international conference on trust in agent societies*, 10 May 2016, vol. 1578, pp. 18–29, https://faculty.cc.gatech.edu/~lingliu/papers/2016/M2MTrust-AAMAS-trust2016.pdf
5. Loper ML. Simulation trust and the internet of things. In: *Proceedings of the 2019 winter simulation conference (WSC)*, National Harbor, MD, 8–11 December 2019, pp. 2–16. New York: IEEE.
6. Rioux GP and Nance RE. Generalizing: is it possible to create all-purpose simulations. In: *Proceedings of the winter simulation conference*, San Diego, CA, 8–11 December 2002, vol. 1, pp. 783–790. New York: IEEE.
7. Henriksen JO. Taming the complexity dragon. *J Simul* 2008; 2: 3–17.
8. Kinder A, Henshaw M and Siemieniuch C. System of systems modelling and simulation–an outlook and open issues. *Int J Syst Syst Eng* 2014; 5: 150–192.
9. Garrett RK Jr, Anderson S, Baron NT, et al. Managing the interstitials, a system of systems framework suited for the ballistic missile defense system. *Syst Eng* 2011; 14: 87–109.
10. Boyd J. *A discourse on winning and losing*. Maxwell Air Force Base, AL: Air University Press, 1987, https://www.airuniversity.af.edu/Portals/10/AUPress/Books/B_0151_Boyd_Discourse_Winning_Losing.pdf
11. Roman E. OUSD(R&E) mission integration overview: 2021-2022 system of systems engineering collaborators information exchange webinars, 2021, https://mitre.tahoe.appsembler.com/asset-v1:mitre+SOS0+open+type@asset+block@08-10-2021-SoSECIE-Roman-OUSD_ME_State_of_Practice.pdf
12. Ladyman J, Lambert J and Wiesner K. What is a complex system? *Eur J Philos Sci* 2013; 3: 33–67.
13. Guichard D. An introduction to combinatorics and graph theory, 2017, https://www.whitman.edu/mathematics/cgt_online/cgt.pdf
14. Urias V, Van Leeuwen B and Richardson B. Supervisory Command and Data Acquisition (SCADA) system cyber security analysis using a live, virtual, and constructive (LVC) testbed. In: *Proceedings of the 2012 IEEE military*

communications conference (MILCOM 2012), 29 October 2012, pp. 1–8, https://www.osti.gov/servlets/purl/1117295
15. Moreland JD. *DoD mission engineering and integration guidebook*. Fort Belvoir, VA: DoD Defense Acquisition University, 2020.
16. Boyd J. *Destruction and creation*. Leavenworth, WA: US Army Command and General Staff College, 1976.
17. Fong B and Spivak DI. *An invitation to applied category theory: seven sketches in compositionality*. Cambridge: Cambridge University Press, 2019.
18. Halter M, Herlihy C and Fairbanks J. A compositional framework for scientific model augmentation, https://arxiv.org/pdf/1907.03536.pdf
19. Censi A. Uncertainty in monotone codesign problems. *IEEE Robot Autom Lett* 2017; 2: 1556–1563.
20. Fairbanks JP, Kannan R, Park H, et al. Behavioral clusters in dynamic graphs. *Parall Comput* 2015; 47: 38–50.
21. Patterson E. Hausdorff and Wasserstein metrics on graphs and other structured data. *Inform Infer* 2021; 10: 1209–1249.
22. Jacobs B, Kissinger A and Zanasi F. Causal inference by string diagram surgery. In: Bojańczyk M and Simpson A (eds) *Foundations of software science and computation structures*, 2019, pp. 313–329. Cham: Springer.
23. Loper M and Garrett RK Jr. *A comparison of traditional simulation and the MSAL approach*. Technical Report ICL-DO-01-15, 2015, Georgia Tech Research Institute Information & Communications Laboratory, https://pdfslide.net/documents/a-comparison-of-traditional-simulation-and-msal-6-3-2015.html
24. Zakaria N. Action network: a probabilistic graphical model for social simulation. *Simulation* 2022; 98: 335–346.
25. Bonabeau E. Agent-based modeling: methods and techniques for simulating human systems. *Proc Natl Acad Sci USA* 2002; 99: 7280–7287.
26. Dempster AP. A generalization of Bayesian inference. *J Roy Stat Soc Ser B Methodol* 1968; 30: 205–232.
27. Shafer G. *A mathematical theory of evidence*. Princeton, NJ: Princeton University Press, 1976.
28. Mabrok MA and Ryan MJ. Category theory as a formal mathematical foundation for model-based systems engineering. *Appl Math Inf Sci* 2017; 11: 43–51.
29. Beheshti R and Sukthankar G. Analyzing agent-based models using category theory. In: *Proceedings of the 2013 IEEE/WIC/ACM international joint conferences on web intelligence (WI) and intelligent agent technologies (IAT)*, Atlanta, GA, 17–20 November 2013, vol. 2, pp. 280–286. New York: IEEE.
30. Henkel R, Wolkenhauer O and Waltemath D. Combining computational models, semantic annotations and simulation experiments in a graph database. *Database* 2015; 2015: bau130.
31. Saleem A, Mahmood I, Sarjoughian H, et al. A water evaluation and planning-based framework for the long-term prediction of urban water demand and supply. *Simulation* 2021; 97: 323–345.
32. City Protocol Society. City anatomy: a framework to support city governance, evaluation and transformation, 2015, https://cityprotocol.cat/wp-content/uploads/2019/07/CPA-I_001-v2_City_Anatomy.pdf
33. Joyal A and Street R. The geometry of tensor calculus, I. *Adv Math* 1991; 88: 55–112.
34. Libkind S, Baas A, Patterson E, et al. Operadic modeling of dynamical systems: mathematics and computation, https://www.cl.cam.ac.uk/events/act2021/papers/ACT_2021_paper_53.pdf
35. Boeing G. OSMnx: new methods for acquiring, constructing, analyzing, and visualizing complex street networks. *Comput Environ Urb Syst* 2017; 65: 126–139.
36. Brooks RT and Sage AP. System of systems integration and test. *Inform Knowl Syst Manage* 2005; 5: 261–280.
37. Luna S, Lopes A, Tao HY, et al. Integration, verification, validation, test, and evaluation (IVVT&E) framework for system of systems (SoS). *Proced Comput Sci* 2013; 20: 298–305.
38. Madni AM and Sievers M. System of systems integration: key considerations and challenges. *Syst Eng* 2014; 17: 330–347.
39. Zeigler BP and Nutaro JJ. Towards a framework for more robust validation and verification of simulation models for systems of systems. *J Defense Model Simul* 2016; 13: 3–16.
40. Felder WN. The U.S. National Airspace System: a model for verification and validation of complex, distributed systems-of-systems. In: *Proceedings of the 16th AIAA aviation technology, integration, and operations conference*, 2016, P. 3152, https://arc.aiaa.org/doi/10.2514/6.2016-3152
41. National Research Council. *Assessing the reliability of complex models: mathematical and statistical foundations of verification, validation, and uncertainty quantification*. Washington, DC: National Academies Press, 2012.
42. Grandison T and Sloman M. A survey of trust in internet applications. *IEEE Commun Surv Tut* 2000; 3: 2–16.
43. Ahn J, DeAngelis D and Barber S. Attitude driven team formation using multi-dimensional trust. In: *Proceedings of the 2007 IEEE/WIC/ACM international conference on intelligent agent technology (IAT'07)*, Fremont, CA, 2–5 November 2007, pp. 229–235. New York: IEEE.
44. Matei I, Baras JS and Jiang T. A composite trust model and its application to collaborative distributed information fusion. In: *Proceedings of the 2009 12th international conference on information fusion*, Seattle, WA, 6–9 July 2009, pp. 1950–1957. New York: IEEE.
45. Voas J, Kuhn R, Laplante P, et al. Internet of Things (IoT) trust concerns, NIST cybersecurity white paper, 2018, https://csrc.nist.gov/CSRC/media/Publications/white-paper/2018/10/17/iot-trust-concerns/draft/documents/iot-trust-concerns-draft.pdf
46. Cyber-Physical Systems Public Working Group. *Framework for cyber-physical systems: volume 2, working group reports*. NIST Special Publication 1500-202, 2017, http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.1500-202.pdf
47. Pinyol I and Sabater-Mir J. Computational trust and reputation models for open multi-agent systems: a review. *Artif Intell Rev* 2013; 40: 1–25.
48. Leister W and Schulz T. Ideas for a trust indicator in the internet of things. In: *Proceedings of the first international conference on smart systems, devices and technologies*, May 2012. Oslo: SMART Press, https://www.researchgate.net/publication/230585169_Ideas_for_a_Trust_Indicator_in_the_Internet_of_Things
49. Terzis S. Trust management. *IEEE Comput Soc Comput Now* 2009; 8: 1296.

50. Yan Z, Zhang P and Vasilakos AV. A survey on trust management for internet of things. *J Netw Comput Appl* 2014; 42: 120–134.
51. Ortt D. An improved methodology for expressing and communicating tropical cyclone risk. In: *Proceedings of the 33rd conference on hurricanes and tropical meteorology*, 17 April 2018, https://www.semanticscholar.org/paper/An-Improved-Methodology-for-Expressing-and-Tropical-Ortt/793942580d749c41e23c8a15674f95ef1dcdc7f2
52. Maher N, Milinski S and Ludwig R. Large ensemble climate model simulations: introduction, overview, and future prospects for utilising multiple types of large ensemble. *Earth Syst Dyn* 2021; 12: 401–418.

Author biographies

Robert K Garrett Jr focuses his attention on adapting, developing, and exploiting modeling, simulation, and analysis tools and processes for ME and integration (MEI) of complex, adaptive SoS. He is now retired from the Defense Department and works part time with Georgia Tech Research Institute working on mathematically rigorous MEI process.

James P Fairbanks is an Assistant Professor in Computer and Information Science and Engineering at the University of Florida. He studies mathematical modeling and scientific computing through the lens of abstract algebra and combinatorics, and leads the AlgebraicJulia.org project. He is has won both the DARPA Young Faculty and Director's awards supporting his work on ACT and scientific computing.

Margaret L Loper is the associate laboratory director and laboratory chief scientist for the Information and Communications Laboratory at the Georgia Tech Research Institute. She has worked in M&S for more than 30 years, specifically focused on parallel and distributed systems. Her current research focuses on edge intelligence and computational trust.

James D Moreland Jr is the Vice President, Mission Integration in the Air Power business unit for Raytheon Technologies in the Missiles and Defense organization. He retired as the Office of the Under Secretary of Defense for Acquisition and Sustainment (OUSD (A&S)) Executive Director, Mission Engineering and Integration (MEI) with 31 years of service. He is a world-renowned expert in MEI and served as a Senior Executive Science Advisor to the White House Office of Science & Technology Policy providing advice on scientific, engineering, and technological matters.