Proceedings of the ASME 2023 International Design Engineering Technical Conferences
Computers and Information in Engineering Conference
IDETC/CIE 2023
August 20-23, Boston, MA

IDETC2023-116689

MODELING DISTRIBUTED SITUATION AWARENESS IN RESILIENCE-BASED DESIGN OF COMPLEX ENGINEERED SYSTEMS

Lukman Irshad[1,*†], Daniel Hulse[2†]

[1]Intelligent Systems Division (KBR, Inc.), NASA Ames Research Center, Moffett Field, CA 94035
[2]Intelligent Systems Division, NASA Ames Research Center, Moffett Field, CA 94035

ABSTRACT

Human operators play a major role in the resilience of complex systems–while human error is one of the biggest contributors to hazardous events, operators additionally play a critical role in mitigating hazardous events. A key factor underlying this operator resilience is situation awareness–the ability of operators to understand their environment and each other to achieve desired system functions. In contrast to situation awareness-related accident models in the literature, which are largely conceptual in nature, this work proposes the use of a dynamic simulation framework to concretely model both the effects of situation awareness-related human errors and situation awareness-related hazard-mitigating properties using the distributed situation awareness theory. This work then presents specialized model constructs to enable agents' individual perceptions of the system state and transactions with other agents (and thus distributed situation awareness) to be represented in simulation. To demonstrate this framework, it is then adapted to an aircraft taxiway case study, where it is used to model aircraft conflicts due to lack of vision and poor communications from the air traffic controller. This demonstration shows the potential of using simulation models to rigorously understand situation awareness-related human errors and thus inform the design of resilience.

Keywords: Resilience Modeling, Distributed Situation Awareness, Complex Systems Design, Simulation

1. INTRODUCTION

Human error is a major contributor to accidents and performance losses in complex engineered systems [1, 2]. If one examines these human-error caused failures further, the largest single contributor to these errors is lack of situation awareness. Studies of aviation accidents involving major air carriers have shown that situation awareness was the root cause of around 90% of accidents involving pilot error [3, 4]. Another study explored offshore drilling accidents involving human error and found that 40% of accidents were directly attributed to the loss of situation awareness [5]. Studies of human errors in other domains such as nuclear power, air traffic control, the process industry, and the automotive industry show that loss of situation awareness is a root cause in a majority of the events [6–8]. Situation awareness-related failures are not only common but also costly and fatal (e.g., Bhopal Gas Leak [9], Air France 447 Flight Crash [10]). Thus, the concept of situation awareness has emerged as an important construct to consider in the engineering process to ensure the system is resilient to potentially-hazardous events.

Resilience is the ability of a system to prevent and mitigate hazardous scenarios, which may be achieved not only by ensuring the system can recover performance after a hazardous event, but by making the system robust to event occurrence, giving the system "graceful extensibility" so it can mitigate surprise events, and ensuring a system adapts to maintain these properties under changing conditions [11]. Since humans often take the role of adapting, responding to, and mitigating hazardous events in complex engineered systems, resilience engineering posits that humans are not solely a threat vector, but are also a key part of maintaining safe operations [12]. Taking this point of view, not only does the lack of situation awareness lead to failures, but (positive) situation awareness should be seen as a key contributor to the operator's ability to mitigate hazardous events as they arise–improving their resilience [13]. Because of this, designing resilient complex engineered systems should involve considering and encouraging agent situation awareness starting from the early phases of the process (i.e., in concept development and requirements formation).

There are three types of situation awareness models: individual situation awareness (e.g., three level situation awareness theory [14]), team situation awareness (e.g., individual vs shared situation awareness theory [15]), and Distributed Situation Awareness (DSA) [16]) models. While individual situa-

†Joint first authors
*Corresponding author: lukman.irshad@nasa.gov
Documentation for `asmeconf.cls`: Version 1.34, May 15, 2023.

tion awareness models can give important insight into individual agents' situation awareness, the situation awareness in a complex system is not merely the sum of individual agents' situation awareness [17]. Also, situation awareness in a system may not always be shared [18, 19]. For example, an air traffic controller and a pilot do not have access to the same system states, but their situation awareness does nevertheless enable them to perform their functional roles in the system. Additionally, as systems become increasingly automated, situation awareness can no longer be considered solely as a human property–instead, it must be considered as a property of all elements in a system that perform control functions (including humans, hardware, and software) [18]. As a result, DSA models are more appropriate for complex systems when compared to individual or team situation awareness models because they take a systemic view of situation awareness and treat it as a collective outcome rather than an individual cognitive process [19, 20]. However, individual situation awareness models can be used to complement the systemic situation awareness models by enabling the study of individual agents' situation awareness [17]. According to the DSA theory [16, 17], situation awareness is a result of the coordination between the elements within a system, and thus a property of the system as a whole. The totality of situation awareness-related information elements, held by and distributed among the agents and artifacts (both human and non-human) of the system, represents the overall system's DSA. While the DSA model has been applied in a variety of industries (e.g., aviation [10], process control [21], military [22] medical [23, 24], sports [25], pandemic response [26], etc.), these applications have only been to either validate the DSA model, study the DSA of existing systems, promote DSA in systems that have been already designed, or investigate past failures. As a result, these approaches have mainly been helpful for improving systems (in terms of information availability, communications, and organizational structure) after they have been implemented, rather than taking a proactive design approach.

In the engineering process, DSA should be considered early on so that operator resilience can be built into the system design (avoiding costly modifications which would need to happen if the design was inadequate [27]). Treating situation awareness as a systemic construct (via the DSA model), rather than an individual cognitive function, can enable this early design stage consideration of situation awareness, since it may be treated as a characteristic of system interfaces (which can be designed) and interactions rather than the fault of individual agents. One of the more prominent approaches for considering human accidents is the Systems Theoretic Accident Model and Process (STAMP), which focuses on considering systemic interactions between operators and their controlled processes, rather than attempting to capture the specific chains of events that cause hazards [28, 29]. The resulting STPA hazard analysis and systems engineering approach has further been developed to prevent human accidents which occur as a result of inadequate control processes [30]. For the specific consideration of DSA, the RiskSOAP [31] method was further developed based on these methods to quantify (and thus improve) metrics related to DSA. While these approaches are helpful for understanding situation awareness-related errors, they are largely conceptual in nature (that is, an analyst must carry out the resulting hazard assessment process) and are thus inadequate for the early design of resilience, which may require analyzing the dynamic behavior of the system over hazardous scenarios across different design options [32].

In this research, we propose the integration of the DSA model into a dynamic simulation of system resilience to hazardous scenarios to enable designers to model the influence of situation awareness on resilience during early design stages. To achieve this, this research will extend the fmdtools toolkit [33] which is being developed to enable the design and analysis of systems resilience using a dynamic systems modeling approach. While fmdtools can currently model human, machine, and software elements of a system and their interactions [34], so far it has not been used to analyze situation awareness-related failures at a system level. While previous work has been conducted to model individual situation awareness failures [35], this work did not address DSA, which requires a more sophisticated analysis to account for the multiagent nature of the situation–where each agent may take a number of roles in contributing to the DSA (e.g., perceiving, storing, processing, communicating, and acting on information). This research will help shift the focus of DSA models from conceptual models to quantitative simulations and enable the iterative assessment of different concepts in early design. Additionally, by approaching DSA errors from a resilience perspective, it will enable the study of not just how lost DSA leads to failures, but of how DSA can prevent externally-caused hazardous scenarios from leading to high-impact failures. To demonstrate this framework, a case study of a semi-automated airport taxiway will be presented and used to compare various approaches and metrics for DSA-based resilience quantification.

2. BACKGROUND

This section details related work in Distributed Situation Awareness (DSA) theory and explores the tools and frameworks that model and quantify DSA. Next, we summarize important context in the development of the fmdtools resilience modeling framework used (and extended) in this work to model human errors and resilience.

2.1 Distributed Situation Awareness

Artman and Garbis [36] were the first ones to take a systemic approach to describe Situation Awareness. Their work was inspired by the the distributed cognition theory [37], according to which cognition is a systemic function that is a result of the coordination between the system agents (both human and technical). Artman and Garbis [36] argued that the system should be viewed as a unit when analyzing situation awareness, and thus described situation awareness as an emergent property of the system rather than it being individual. Stanton et al. [16] proposed the Distributed Situation Awareness (DSA) model, which follows on Artman and Garbis's [36] work. The DSA theory is built on eight core tenets [18], which argue that situation awareness is an emergent behavior of a system and it can hold loosely coupled systems together. The system's situation awareness can be represented by a network of information elements. Each agent within the system have a different view of this information network, and these views are dynamic and can change over time based on the

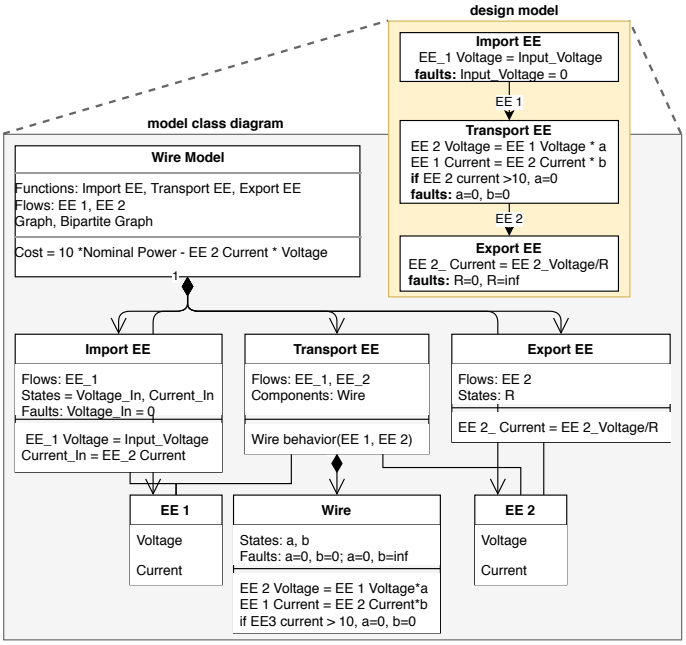

FIGURE 1: OBJECT-ORIENTED MODELING FRAMEWORK IN FMDTOOLS. FROM [33]

goals and tasks performed. Additionally, the situation awareness related information (knowledge) is owned, shared, or used by the agents based on the goals of the tasks performed. Therefore, agents may have different situation awareness for the same situation, but their situation awareness will be compatible. The exchanges (e.g., communications) between agents sustain DSA and as a result, one agent can compensate for another agents degraded situation awareness. This view to situation awareness has allowed the shift from situation awareness being a purely cognitive construct to a systemic construct, which allows the consideration of situation awareness beyond just humans (e.g., situation awareness of Autonomous vehicles) [18].

The Event Analysis of Systemic Teamwork (EAST) [38] is the most commonly used method to model and analyze DSA. It lays out a systematic process to gather DSA-related information (via task analysis, expert interviews, usage logs, observational studies, or communication logs) and generate three types of networks; information network, task network, and social network. Then, a composite network that combines the three types may be used to study DSA further. A specific type of network–propositional network–is created in this process to capture DSA-related knowledge effectively. These networks are used to understand DSA qualitatively and quantitatively via network theory (e.g., number of connections [39], broken nodes [40], broken links [41]). Other types of networks (e.g., concept maps [20], Bayesian Belief Networks [42] have also been used to model DSA. All of the above methods focus solely on DSA, giving minimal attention to the interactions between DSA and other types of vulnerabilities in the system and their propagation effects. This approach is insufficient for design, however, since it lacks a way to concretely map DSA errors onto credible measures of overall risk that can be traded against other design considerations. Taking a different approach, RiskSOAP [31] was developed to quantify risk-related situation awareness in DSA modeling. This method quantifies the difference between the systems' ideal risk perception expectation versus the current risk perception capabilities. It uses a hazard analysis and an early warning hazard assessment to identify the ideal and as-is system risk perception. While this approach accounts for the downstream effects of DSA-related issues by comparing the end state of the ideal versus the real system, it only focuses on the DSA related to potential risk. In this research, we address these limitations by enabling the modeling of DSA in a dynamic simulation of system behavior so the downstream effects of DSA (in terms of system performance and human contribution to (or the lack of) DSA can be studied.

2.2 Resilience Modeling

Resilience modeling is the practice of simulating how a system performs under hazardous scenarios over time to enable the use of a resilience framework/metrics such as the resilience triangle [43]. To enable the assessment of system resilience, the fmdtools toolkit [1] [33] works by simulating the dynamic effects of hazardous modes and conditions in a functional model of the system. This functional model is composed of two types of nodes: functions (which represent the behaviors of the system and their associated data structures, e.g., components or activities) and flows (which represent shared variables between the functions e.g., the flow of materials, energy, and signals). Flows enable the propagation of faulty behaviors between subsystems, however, they also pose a limitation to representing communications/perception between different agents, since functions in this framework fundamentally share the same copies of variables. This representation enables the dynamic simulation of system resilience that takes a high-level functional view of the system behaves so that resilience can be considered in early design, before the system components have been designed in detail.

Recently, to enable the representation of human resilience in complex systems, the fmdtools modeling approach has been extended with a number of different human-related constructs, including human error probability models and action sequence graphs [35]. Action sequence graphs represent the sequence of steps needed by a function (or agent/human) to complete a task in a sequence. These action sequence graphs can be used to represent individual cognitive constructs (e.g., individual situation awareness model [14]) or team dynamic constructs (e.g., team situation awareness model [15]). However, this representation is not capable of representing the systemic cognitive constructs (e.g., DSA) alone, because it lacks a sufficient representation of the underlying interactions between agents and the environment. While the existing flow implementation (in which functions/agents share variables) is appropriate for modeling tightly coupled behaviors (e.g., energy/force/material flows), it is unwieldy to use for signal flows that constitute messages which must be passed back and forth between agents. The objective of this research is to overcome this limitation by augmenting the fmdtools framework so that it can readily represent not just agent actions, but the flow of information between agents via communications and thus

[1] https://github.com/nasa/fmdtools

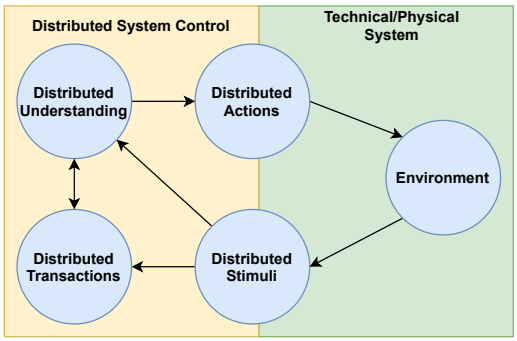

FIGURE 2: HIGH-LEVEL FRAMEWORK FOR REPRESENTING DISTRIBUTED SITUATION AWARENESS IN SIMULATION

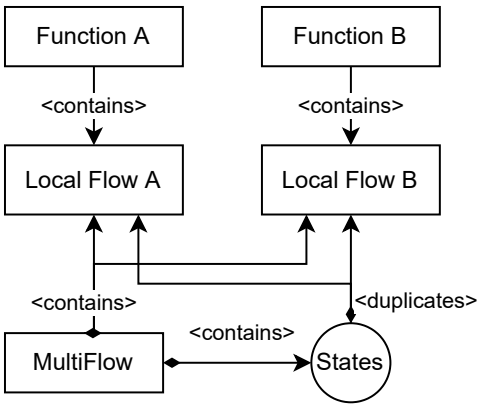

FIGURE 3: MULTIFLOW STRUCTURE FOR REPRESENTING DISTRIBUTED PERCEPTION

advance the state of the art in resilience modeling to include DSA-related resilience considerations.

3. METHODOLOGY

To enable the holistic consideration of Distributed Situation Awareness (DSA) in systems resilience analysis, this section presents a framework to represent DSA properties in resilience simulation models. However, it is first important to understand how DSA can be represented at a system level so that the related behaviors (e.g., the process of perception and communications) can be represented in detail. The systemic representation of DSA is captured in this work via the system functional model that embodies the framework shown in Fig. 2, where controllers take actions in an environment based on stimuli and their transactions (or communications) with each other. This is in essence a STAMP model [28], except that the underlying agent process models and control algorithms are distributed to different agents which may perform different functional roles in the system. The systemic representation of DSA is in turn captured in this framework via the system functional model by using new flow types which represent the situation awareness transactions. Situation awareness transactions are the exchange of information to and/or from agents and can be broken down into two main types–distributed transactions, which are exchanges between agents (e.g., communications), and distributed stimuli, which are stimuli from the environment perceived by an agent (e.g., what an operator sees). In other words, the system's situation awareness is composed of stimuli and exchanges. Individual agents have different views of these stimuli and exchanges, which make them distributed in nature (i.e., Distributed Stimuli and Distributed Transactions of agents). The collection of Distributed Stimuli and Distributed Transactions of an agent will form the situation awareness of that agent. However, with the exchanges and stimuli, they still need to be perceived, comprehended, and acted upon which we refer to as Distributed Understanding and Distributed Actions, as shown in Figure 2.

The next subsection describes how these properties may be represented for the simulation of DSA-related faults for resilience assessment.

3.1 Distributed Stimuli

Distributed Stimuli is the agent's view of a given environmental (controlled or otherwise) phenomenon. That is, each agent has its own independent view of the environment based on its role, goals, tasks performed, configuration, and performance shaping factors (in case of a human agent). Agents may individually perceive their environment (e.g., by their senses or instruments). This view of the environment is in turn limited by the scope of the measurements (i.e., what is being measured), accuracy of the perception, as well as the internal bandwidth (e.g., mental capacity) of the agent. Perceiving Distributed Stimuli has two major considerations for the modeling of resilience: the inclusion of fault modes and behaviors. Fault modes are sources of hazardous scenarios, and constitute potential ways a stimuli may become unavailable or ways an agent might fail at perceiving the stimuli. For example, an autonomous rover may not be able to gauge its position if the video feed is broken (loss of stimuli). Examples of failed perception include an agent failing to perceive an object or perceiving the wrong number on a gauge. The second consideration, behaviors, constitute how the agent gathers and fuses this information, and thus whether the perception fault leads to further hazardous consequences (which may also be considered an aspect of Distributed Understanding if it requires cognition). In this work, the `MultiFlow` class was developed for representing Distributed Stimuli tasks, as shown in 3. This class enables agents to each have their own local views of the same states, and update these views from their "true" values. Faults related to perceiving Distributed Stimuli may then be implemented by modifying how local states are updated from their true values (e.g., by not updating or updating with wrong/modified values).

3.2 Distributed Transactions

Distributed Transactions is the view an agent has to information Transactions (e.g., communications, sensor data, situation awareness, etc.) that occur in the system. Distributed Transactions can be thought of as an extension of the Distributed Stimuli concept in which states are passed between agents rather than between agents and the environment. As a result, the Distributed Transactions is subject to all of the limitations, faults, and behaviors which affect Distributed Stimuli. However, when representing transactions, faults can occur on the side of the sender, on the side of the receiver, or both. Additionally, behavioral resilience can be represented both in the behaviors of the individual agents

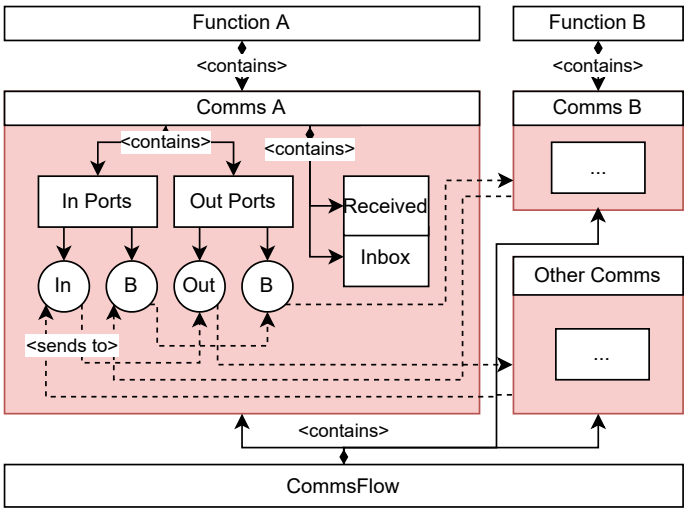

FIGURE 4: COMMSFLOW STRUCTURE FOR REPRESENTING DISTRIBUTED TRANSACTIONS

(update interval, etc.) but also in the overall transaction structure (e.g., hub and spoke vs decentralized communications).

In this work, the `CommsFlow` class was developed to represent the transfer of information for Distributed Transactions, as shown in Fig. 4. This class extends the `MultiFlow` class via the incorporation of in states and out states, which enable the passing of situation awareness transactions, either via explicit ports (carrying transactions to/from a specific agent) or via the generic in/out port (which carries transactions to/from all other agents). The passing of transactions is then managed by "inbox" and "recieved" data structures which specify whether a transaction is currently being sent between two agents, or has already been read (thus clearing the way for another transaction). The carrying of transactions is then used by "send" and "recieve" methods, where sending (on the side of the sender) updates the corresponding out port for the transaction, and updates the inbox of the corresponding receiver, and receiving (on the side of the receiver) updates the corresponding port for the state and carries the transaction receipt from "inbox" to "received" to enable new transactions to be sent to the receiver. Distributed Transactions faults can then be implemented by modifying how these messages are passed between agents (as with Distributed Stimuli) on the side of the sender or receiver, as well as by modifying the inbox/received properties of the local `CommsFlow` of the agent.

3.3 Distributed Understanding

Distributed Understanding is the property of agents to abstract meaningful (as in, actionable) information based on their situation awareness transactions and stimuli. Since agents are faced with multiple heterogeneous (and possibly conflicting) situation awareness transactions and stimuli, they must in turn make sense of these stimuli and transactions in order to act. This is limited by mental capacity and reasoning capabilities of the agents. As such, modeling the Distributed Understanding property involves modeling the agents' internal information processing, reasoning, and decision-making characteristics. The process of Distributed Understanding may be different for human and non-human agents. For instance, for humans, this may involve cognitive reasoning that depends on factors such as performance shaping factors, tasks performed, goals, and mental model, whereas for a neural network-based autonomous agent, it may depend on factors such as data quality, training method, and processing capacity.

As a simulation construct that represents human reasoning, distributed understanding can be modeled using (1) internal states (reflecting the agents' internal understanding of itself or its environment) and (2) agent behaviors reflecting the agent's decision-making and reasoning (e.g., projection, problem-solving consideration of uncertainty, etc.). The first component may be represented via an "internal" version of the `MultiFlow` or `CommsFlow` constructs developed in the previous subsections. Behaviors of internal states of an agent may be represented by considering how often the states in the flows are updated, the number of states updated, how the states are processed, and how factors such as performance shaping factors affect behaviors. For example, for the perception of Distributed Stimuli, checking a measurement frequently/infrequently or the number of redundant sources of information used can be modeled by altering the internal state of `MultiFlow` and related behavioral faults may be induced through the same. For Distributed Transactions, behavioral resilience can be represented (in addition to updating) via the communications structure (e.g., ports) and the way transactions are sent/received by the agents in the `CommsFlow`.

The second component of Distributed Understanding is more difficult to represent, since it may be task/scenario and agent dependent. In this work, this part is represented as an integral component of the agent behaviors, which may be embodied in simulations as state machines, action sequence graphs, if/else statements, or a sequence of function calls. While past work [17] has proposed a DSA model (via Schema theory and Perceptual Cycle) that model this understanding in humans, there is very little details in terms of how DSA will be gained and acted upon in autonomous agents. One is not limited to DSA related models when choosing cognitive processing models. For simulation purposes, designers may choose any cognitive reasoning model (e.g., Three Level Situation Awareness Model [14]) as they see fit depending on their application. If the modeling is done at a high level, the cognitive models can be converted to work for non-human agents to make the Distributed Understanding related simulations possible for non-human agents. For example, using the three level situation awareness theory at a high level, agents have to perceive, comprehend, project, and act upon the situation awareness-related information regardless of them being a human or non-human agent. This perspective may be adopted to model Distributed Understanding in non-human agents. This high level modeling is especially more feasible during early design stages when very little system information is available.

3.4 Physical Effects and Metrics

Finally, DSA can be measured by its effects on the agents and their environment via actions made by the agents as shown on the right side of Fig. 2. While not explicitly a part of DSA, Distributed Actions determine the effects which result from the agents' DSA, which is a function of agent roles and capabilities. Modeling these

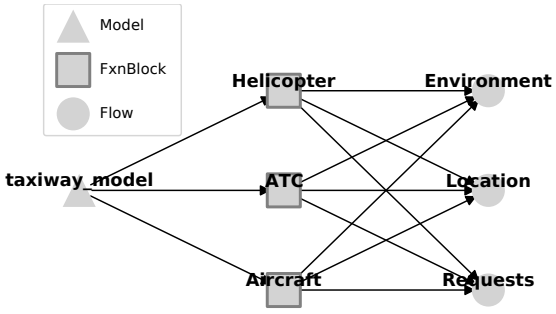

FIGURE 5: MODEL CLASS STRUCTURE

effects enables the consideration of the external effects of DSA faults, as well as how feedback between the environment and agent DSA, understanding, and actions unfolds over time. As a result, these effects are necessary both for the understanding of severity (i.e., external consequences of DSA faults) and resilience (i.e., how DSA faults unfold over time). Based on these effects, severity metrics can thus be identified (which have been developed in previous work [33]) and calculated for the system to quantify the risks associated with scenario consequences.

Aside from severity metrics, the impact of scenarios on DSA can be quantified with their own specialized metrics. In this work, the concept of a "degraded field" is used to developed an indicator of DSA. A degraded field is a model state that differs between the nominal scenario and a faulty scenario under test at a particular timestep, and has been used previously in fmdtools models to highlight the propagation of faults through the graph. The degraded field may be calculated as the indicator function f_d of the nominal value of the field f_n and fault scenario value of the field f_s:

$$f_d = \mathbb{1}_{f_n \neq f_s} \tag{1}$$

where a f_d of 1 means the field has degraded, while an f_d of 0 means the field is nominal. The total percent degradation p_d of the system over n fields in a system can thus be calculated:

$$p_d = 100\% \frac{\sum_i^n f_{d,i}}{n} \tag{2}$$

In this work, the number of these degraded fields is tallied up and calculated as a basis of DSA impact, under the rationale that perceived distributed stimuli and transactions (which are fields in the model) which differ from the nominal state are likely to be incorrect, either because they are themselves a directly result of a faulty perception or communications, or may be a result of faulty transaction being passed through the system. This metric will thus be shown and evaluated Section 4.

4. DEMONSTRATION

In this research, we use an aircraft taxiway model to demonstrate the modeling of Distributed Situation Awareness (DSA) using the constructs presented in Section 3. Using this case study, we plan to analyze if DSA can be modeled and measured during the early design stages, specifically studying how simulating DSA errors can inform our understanding of resilience in complex systems design. We also analyze the effects of DSA errors on performance over time (or, resilience) to understand how

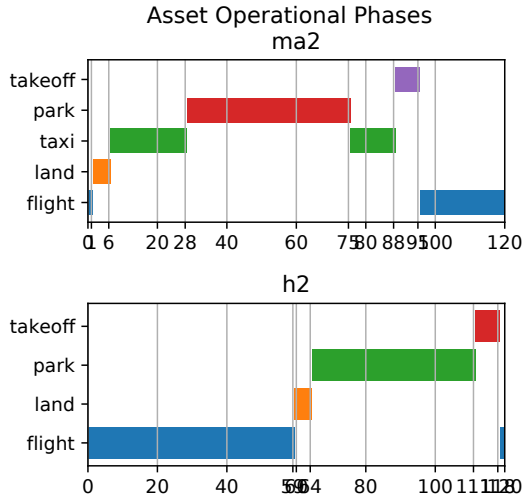

FIGURE 6: AIRCRAFT/HELICOPTER PROGRESSION THROUGH MODES

DSA-related errors propagate into system-level failures. The next sub-sections introduce the model, demonstrate two DSA-related fault scenarios simulated in this model, and then discuss the findings from modeling these scenarios.

4.1 Model Overview

The aircraft taxiway model is composed of `Aircraft`, `Helicopter`, and air traffic control (`ATC`) functions, as shown in Fig. 5, along with `Ground`, `Location`, and `Requests` flows. To describe each function/flow in detail:

- **Aircraft** land in the landing area, taxi to a gate, park at a gate for a defined amount of time, and then taxi to the runway and takeoff. This cycle is shown in Figure 6, which shows the progression of an aircraft through different modes, as they circulate through the taxiway. Aircraft can be additionally be piloted (displayed as `MA` in the figures) or unpiloted (displayed as `UAV` in the figures)–the main (modeled) difference being the size and shape of the vision coverage areas. The vision coverage areas assume a one kilometer visibility. The vision coverage for an unpiloted aircraft is a circular radius, resembling perception from sensors whereas for piloted aircraft, they are a vision cone covering acute, peripheral, and temporal regions.
- **Helicopters** land, park, and take off at the helipad shown in Fig. 7. As such, they can be thought of as a related class to the aircraft that does not taxi. Helipads can only be occupied by one helicopter at a time, or there will be a crash.
- The **Air Traffic Controller** (ATC) assigns areas for takeoff/landing, routes to/from the gate to aircraft, and gives assets the go-ahead to takeoff/land/taxi when they request it.
- The **Ground** flow contains the map shown in Fig. 7, along with three constructs: `area_allocation`, a dictionary that tracks map areas an asset may be present in the map given its instructions (e.g., a taxiing asset would be allocated both to a gate and runway), `asset_area`, a dictionary that tracks

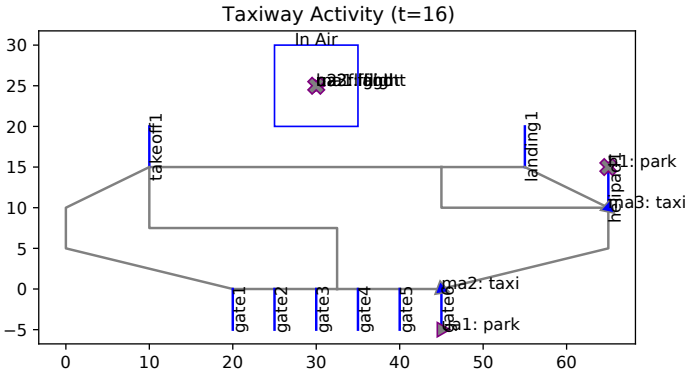

FIGURE 7: MODEL ACTIVITY IN AN EXAMPLE SCENARIO

where assets are on the map, and `asset_assignment`, a dictionary that tracks where each asset is intended to go. As a `MultiFlow`, this flow contains two copies as shown in Figure 13–the "true" version that reflects what each asset is doing, and the ATC version that reflects ATC's perception of what the assets are doing.

- The **Location** flow contains position, velocity, and mode information for each asset. This flow uses the `MultiFlow` class to represent asset location as shown in Fig. 8. In this structure, each asset has a true position (reflecting the asset's true state), a perceived position (reflecting the asset's perception of its state) and a closest position (representing the asset's perception of the closest asset). This closest asset position is determined by the asset by scanning the vision coverage area in front of it, and thus can be updated from all other asset positions while being subject to vision-related perception faults.

- The **Requests** flow contains the desired action to be undertaken by the asset, the clearance by the ATC, as well as the route assignment given by the ATC. This flow uses the `CommsFlow` class to represent the communications between the ATC and assets, as shown in Fig. 12. As shown, this flow has a hub-and-spoke structure where each asset communicates solely with the ATC (though more communications structures are possible with the `CommsFlow` class).

When these functions and flows are instantiated and simulated in a model, it results in the procession of aircraft and helicopter on the taxiway shown in Fig. 7. In this environment, there are six gates, one dedicated landing runway and takeoff runway, and one helipad, along with a few segments between the runways and gates which may be used as routes. Aircraft/helicopters (referred to as assets, a class they both inherit from) either begin the simulation in air, or are randomly assigned a gate or runway in the map. While different parameters (e.g., number of aircraft, park/land time, etc) can be changed to represent different scenarios, this study uses the following parameters. The simulation progresses for 120 timesteps, where each timestep represents one minute. In the nominal scenario, each asset in this simulation (2 helicopters, 3 piloted aircraft, 3 UAVs) should be able to successfully land, park, and takeoff without any accidents, however, this

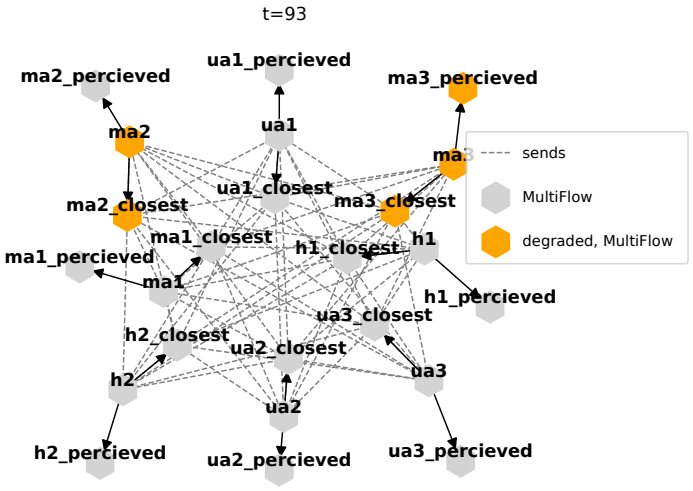

FIGURE 8: VISION FAULT LEADS TO DEGRADATION OF LOCATION PERCEPTION

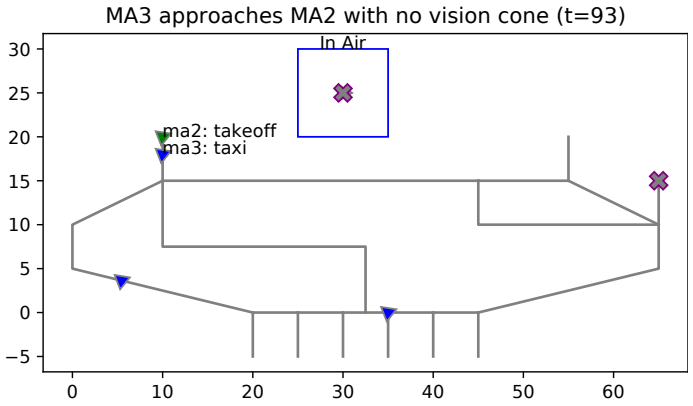

FIGURE 9: THE VISION FAULT CAUSES THE AIRCRAFT TO CRASH AT THE RUNWAY

performance may be modified by fault scenarios (or simulating under other parameters, e.g., more aircraft). The next subsections show the performance of this model in two DSA-related fault scenarios.

4.2 Scenario 1: Aircraft Vision Reduced

To demonstrate the use of this methodology to model the effects of perception faults, in this section the scenario of reduced aircraft vision is considered. In this scenario, the pilot's vision coverage is reduced such that they cannot perceive the closest aircraft on the taxiway, resulting in loss of DSA. However, they are still able to navigate the runway through other means. To demonstrate this scenario, the fault was injected at t=1 in the `ma3` aircraft. This fault leads to no immediate hazardous consequences, since the aircraft is not in close contact with other aircraft until it is queuing to takeoff behind the aircraft `ma2` at t=93, as shown in Fig. 9. At this point, as shown in Fig. 8, the asset `ma3`'s `closest location` flow is degraded from nominal, causing it to degrade its own `perceived location`, as well as `ma2`'s `closest location` flow (and thus it is containing true location). This causes it to crash into `ma2` from behind, as shown

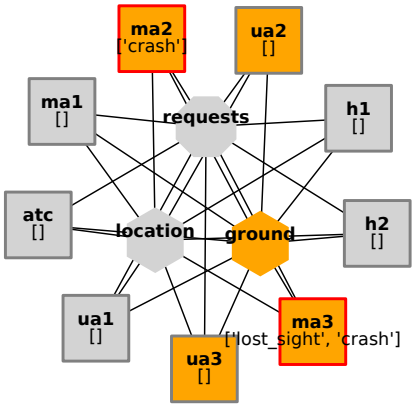

FIGURE 10: RESULTING CRASHES FROM VISION FAULTS

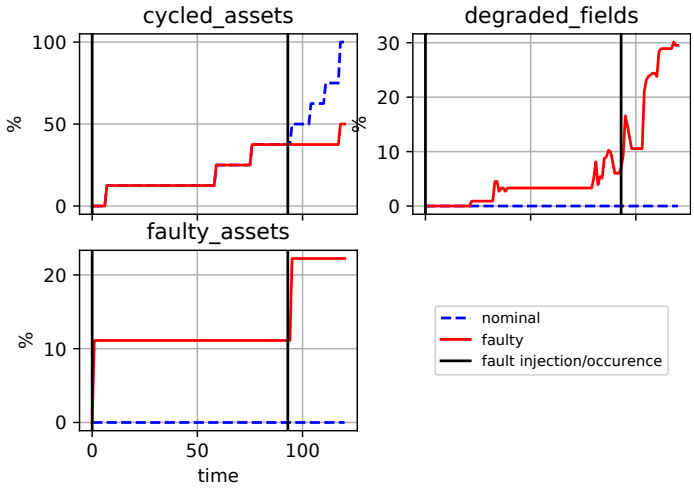

FIGURE 11: PROGRESSION OF VISION FAULT SCENARIO

in Fig. 9. As shown in Fig. 10, ma3's loss of DSA relating to its closest asset also results in the degradation of operations of aircraft ua2 and ua3, which are unable to takeoff and complete their mission due to the blocked runway.

The progression of this simulation is shown in Fig. 11, where degraded fields means the number of model states that were in a degraded state, cycle assets means the number of assets that were activated, and faulty assets means the number of assets that were in a fault mode. As shown, the vision fault injected at t=1 causes a small percentage of fields to degrade from their nominal values (mostly those localized/related to the ma3 asset). This changes when the crash occurs at t=93, resulting in an additional faulty asset, as well as a large increase in degraded fields, since both assets are now immobilized and no aircraft can take off. This results in a reduction in cycled assets after that point, since only the helicopters can take off. The results of this scenario at the final timestep are shown in Table 1. As shown, despite the fault only directly causing an additional fault in the other aircraft because of the crash, it still results in half of the assets not being able to complete their mission, even though only 1/3 of the fields are degraded. This shows how loss of DSA in an individual asset can result in non-localized system-level consequences, even if the rest of the system is functioning properly.

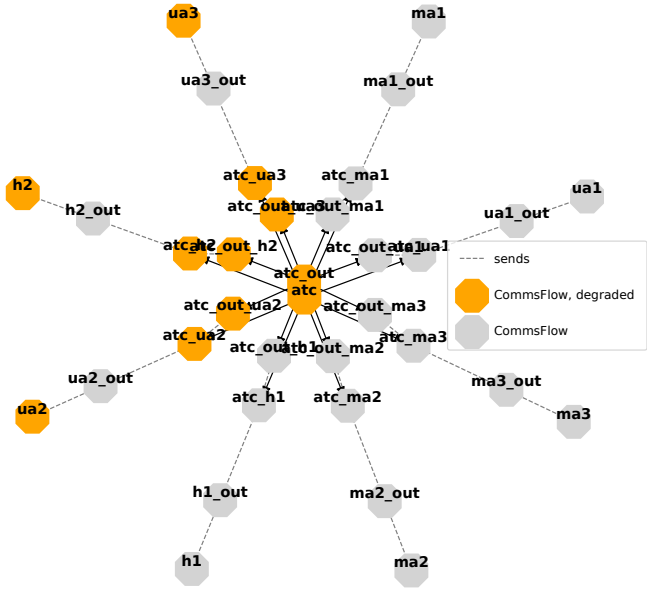

FIGURE 12: INCORRECT COMMUNICATIONS (REQUESTS) RESULTING FROM THE CLEARANCE FAULT (T=10)

TABLE 1: FINAL SUMMARY METRICS FOR VISION FAULT

cycled assets (%)	degraded fields (%)	faulty assets (%)
50.0	29.518072	22.222222

This demonstration further shows how failures in perception can be represented in a simulation model. In this case, a specific perception behavior (perceiving the closest asset location within a vision coverage zone) was modified, resulting in poor actions (getting too close to the asset in front of it and thus crashing). This in turn affects the environment, and thus the perceptions, understanding, and actions of the other agents (not being able to take off), in line with the DSA framework in Figure 2 without an explicit communications component. This limitation in communications is present in that the driving behaviors assume the aircraft rely on line of sight to navigate along set routes and there is no communications between aircraft or the ATC to intervene if they detect aircraft getting too close. This was done to demonstrate perception faults individually–to better show the full framework, the next scenario studies a combined perception/communications fault.

4.3 Scenario 2: ATC Inappropriately Clears Landing

To demonstrate the simulation of communication-based DSA-related faults, this section showcases a scenario in which ATC inappropriately clears aircraft for landing. In this scenario, the ATC incorrectly directs an aircraft to land when there is an aircraft on the runway. This is similar to the "near miss" event that happened in February 2023 in which a FedEx cargo plane was cleared to land as a Southwest flight was set to take off. In this event, the cargo plane was able to pull up at the last minute, averting catastrophic consequences [44]. However, it would have

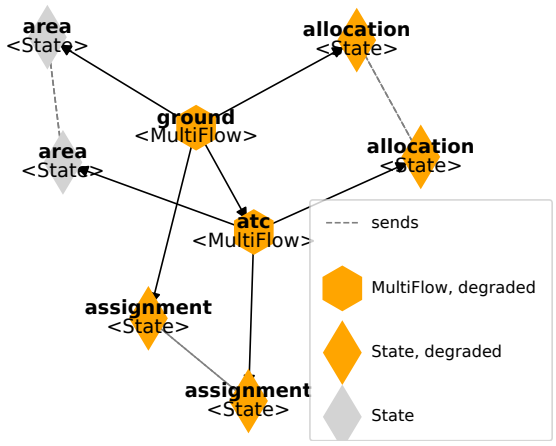

FIGURE 13: INCORRECT AREA ALLOCATION/ASSET ASSIGNMENT RESULTING FROM THE CLEARANCE FAULT (T=11))

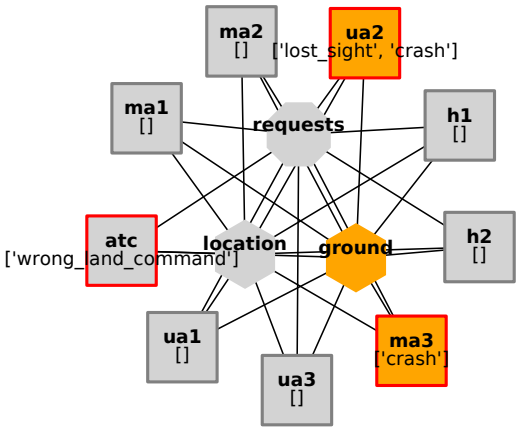

FIGURE 14: RESULTING CRASHES FROM CLEARANCE FAULT

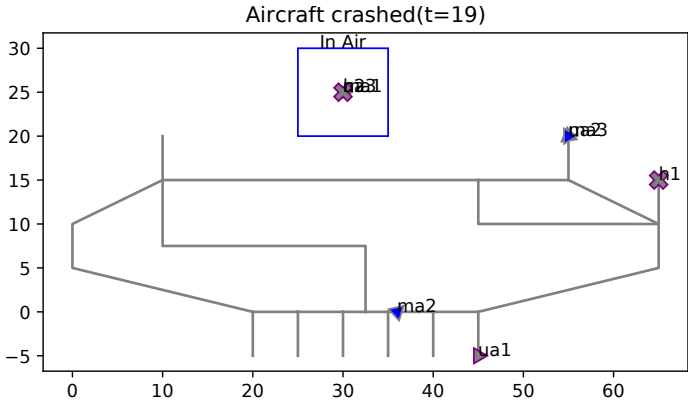

FIGURE 15: RESULTING CLEARANCE FAULT ASSET ALLOCATION–THREE AIRCRAFT ARE INCORRECTLY PERMITTED THE LANDING RUNWAY.

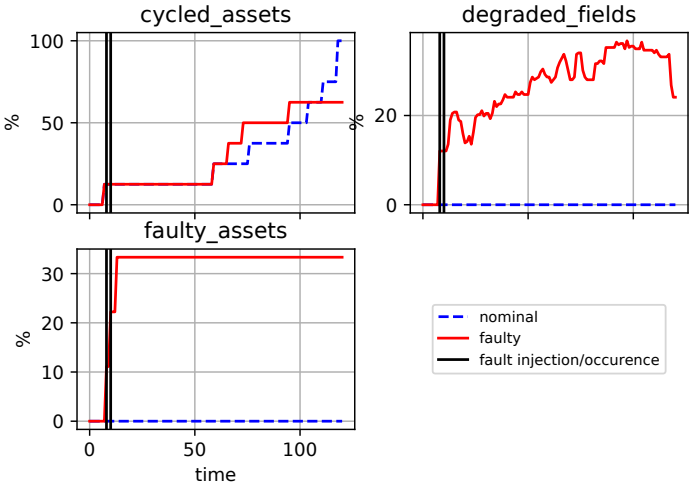

FIGURE 16: PROGRESSION OF CLEARANCE FAULT SCENARIO

been foreseeable for the pilots to not perceive the aircraft on the runway in time. To model this potential worst-case outcome of the near miss event, we simulate the joint fault in which the descending aircraft does not perceive the aircraft on the runway, resulting in loss of DSA.

This joint fault scenario was modeled by first injecting a `wrong land command` fault in the `atc` function at time t=8, followed by a `lost sight` fault in the `ua2` asset at time t=10, during the time when aircraft (`ma3`) is currently taxiing on the runway after just landing. The wrong land command is reflected in the incorrect requests at time t=10, as shown in Fig. 12, which propagate to the aircraft currently in the air (`ua2`, `h2`, and `ua3`). It further results in modified area allocations and asset assignments, as shown in Fig. 13. However, while these assets are cleared to land, only `ua2` attempts a landing, because of its `lost sight` fault, which prevents it from seeing the aircraft `ma3` on the runway. In other words, even though the `wrong land command` caused the

DSA of the assets in the air to degrade, most of them (except for the one with additional DSA degradation's) were able to avoid the crash, because they were able to gain DSA through other stimuli (visual inspection) in the environment. The crashed aircraft was not able to gain situation awareness because of the lack of vision.

This results in the crashes shown in the respective assets, as shown in Fig. 13, and Fig. 15, which occurs because `ua2` lands when `ma3` is still taxing on the runway. The sequence of the fault events and metrics is then shown in Fig. 16. As shown, the faults injected early in the simulation cause a single additional faulty asset (`ma3`) due to the crash, resulting in 33% of the modeled assets being faulty (the atc, along with the two ground assets). This also leads to a high number of degraded fields immediately after the scenario, since these aircraft cannot land. Nevertheless, aircraft continue to cycle, since the takeoff runway is clear and most of the aircraft have already landed and are thus not blocked from taking off. The results at the final timestep are shown in Table 2.

This scenario demonstrates how SA transactions related errors can be modeled in simulation. In this case, communications faults send incorrect information to agents, which in turn either correct for this information (by avoiding the runway at the last minute) or, in the case when they cannot correct for this information (by having inadequate perception), take faulty actions (land at the wrong time). It is also a case that further strengthens the

TABLE 2: FINAL SUMMARY METRICS FOR ATC CLEARANCE FAULT

cycled assets (%)	degraded fields (%)	faulty assets (%)
62.5	24.096386	33.333333

idea that simulation can be used to represent DSA for the assessment of system resilience. Often, resilience is seen as a property of human-operated systems that can mitigate the many potential errors which may come up in operations–ensuring they become "near misses" instead of catastrophic failures [12]. As demonstrated here, simulation can be used to represent what would happen with or without human behavior that promotes system resilience (e.g., seeing the runway or not seeing the runway in time).

4.4 Discussion

This demonstration showed how Distributed Situation Awareness can be simulated in the context of hazardous scenarios for the quantification of human resilience using the constructs introduced in Section 3. The first scenario showed how poor perception (and the resulting DSA degradation) can lead to hazardous actions (and thus consequences) in a situation where Distributed Transactions and Distributed Understanding are not able to provide checks on information. The second scenario showed a situation where a communications error can combine with a perception error to lead to hazardous actions. As shown, both simulations give results which would be expected from their given scenarios, despite being modeled at a high level of abstraction. We thus think there is significant merit in using simulations to quantify hazards and resilience related to DSA.

In this work, a variety of metrics were used to characterize both scenario severity (cycled assets, faulty assets) and DSA impact (degraded fields). One of the main differences between these metrics is their relative severity and impact–that is, degraded fields/cycled assets are likely to vary significantly (and this impacts a large number of assets), their overall severity is low. Faulty assets, in contrast, is relatively bounded to two to three assets which are likely to communicate poorly and/or crash, however it is a high severity metric, since it affects pilot and passenger safety. Another main difference between these metrics is that severity metric characterizes how bad an outcome is, while the DSA-related metric (degraded fields) characterizes the change in DSA between the nominal scenario and faulty scenario. In general, degraded fields is a weak indicator of DSA, since it merely quantifies whether states in a given condition have changed, rather than explicitly calculating which fields are modified in error (e.g., because their agent is faulty) and which field are being modified correctly (i.e., because their agent is correctly perceiving a changed environment). Future work should develop and implement more sophisticated metrics for quantifying DSA errors which take into account intended and unintended modified behaviors.

However, this demonstration also highlights some of the limitations of this framework in understanding the "distributed understanding" component of DSA. In general, operators have more sophisticated capabilities to fuse multiple sources of data to recognize and correct for errors, which is a major contributor to human resilience. Poor human understanding (i.e., inadequate mental models) can also contribute to human errors by causing operators to take inappropriate actions which they thought were appropriate. Thus, while the taxiway model has (simple) human decision-making components, to fully develop a human resilience assessment that fully embodies the overall framework in Fig. 2, future work should investigate and develop constructs and/or a formalism specifically for representing the human decision-making component. One way to achieve this is to implement cognitive reasoning models (e.g., Perceptual cycle theory, three level situation awareness model, etc.) to model the distributed understanding constructs. Nevertheless, the usefulness of this work without these components should be apparent from the demonstration in Section 4–specifically, it enables us to represent the potential (realistic) propagation and effects of DSA-related errors and track the progression of DSA over time.

5. CONCLUSION

To conclude, distributed situation awareness is a key aspect of systems resilience which can (when distributed situation awareness is weak) lead to hazards or (when distributed situation awareness is strong) prevent them from occurring. Situation awareness stimuli and transactions are furthermore key components of distributed situation awareness. To readily enable the consideration of distributed situation awareness for simulations of system resilience, this paper developed specific modeling constructs for representing the situation awareness transactions between agents and their environment in which each agent is able to have copies of system state representing its own awareness of the system, as well as messages to and from other agents. When integrated in the context of an overall system model, this enables the propagation of distributed situation awareness-related faults and the quantification of related hazards and distributed situation awareness degradation.

This framework was demonstrated in the context of a model of aircraft and helicopters navigating an airport taxiway by the direction of air traffic control. As shown in Section 4, this methodology was helpful for understanding how lack of awareness (from degraded perception, communications, or understanding) results in hazards, either in terms of direct consequences, or in terms of consequences which could occur in joint failure scenarios. Because of the high-level of abstraction and simplicity of the underlying model, using this tool can shift the application of distributed situation awareness modeling from later design stages to earlier design stage. As a result, the simulation can be used to proactively design the system (to form requirements) rather than reconfigure the system afterward using operational data. Furthermore, because it is based on an underlying simulation, it enables the iterative testing of different distributed situation awareness strategies and assumptions, rather than relying on designer knowledge alone. This can lead to a better understanding of system vulnerabilities early on, allowing designers to build distributed situation awareness error mitigation into the system rather than treating it as an afterthought. Additionally, the ability to model distributed situation awareness-related faults along

with other types of faults sets this frameworks apart from existing distributed situation awareness modeling frameworks. This will allow designers to analyze how the different types of faults interact and if distributed situation awareness can help mitigate any of the other types of faults. This framework can thus enable designers to build more resilience systems that are less prone to accidents.

The main limitation with this methodology is the representation of agent (both human and non-human) understanding in the context of distributed awareness. In the future, we would like to add further constructs to represent this piece of distributed situation awareness at a greater level of specificity and fidelity. We would also like to demonstrate this scenario in a complete human behavioral modeling framework with human actions represented as Action Sequence Graphs, as has been done in the past [35]. Additionally, as was noted in Section 4.4, very few distributed situation awareness-specific measures have been developed in this model, and those that have (field degradation) have major limitations. Future work should thus enable the modeling of a number of different metrics in this framework to fully represent the degradation of distributed situation awareness. Finally, to better demonstrate the usefulness of this approach in identifying hazards, we would like to explore the use of hazard sampling techniques to automatically create a list of potential ditributed situation awareness-related hazards.

ACKNOWLEDGMENTS

This research was funded by the System-Wide Safety project in the NASA Aeronautics Research Mission Directorate. The findings herein represent the research of the authors and do not necessarily the view of the United States Government or NASA. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement by the United States Government.

REFERENCES

[1] Kohn, Linda T, Corrigan, Janet M, Donaldson, Molla S et al. "Errors in health care: a leading cause of death and injury." (2000).

[2] Högberg, Lars. "Root causes and impacts of severe accidents at large nuclear power plants." *Ambio* Vol. 42 No. 3 (2013): pp. 267–284.

[3] Endsley, Mica R. "A taxonomy of situation awareness errors." *Human factors in aviation operations* Vol. 3 No. 2 (1995): pp. 287–292.

[4] Jones, Debra G and Endsley, Mica R. "Sources of situation awareness errors in aviation." *Aviation, space, and environmental medicine* (1996).

[5] Sneddon, Anne, Mearns, Kathryn and Flin, Rhona. "Situation awareness and safety in offshore drill crews." *Cognition, Technology & Work* Vol. 8 No. 4 (2006): pp. 255–267.

[6] Endsley, Mica R. "Designing for situation awareness in complex systems." *Proceedings of the Second International Workshop on symbiosis of humans, artifacts and environment*: pp. 1–14. 2001.

[7] Naderpour, Mohsen, Nazir, Salman and Lu, Jie. "The role of situation awareness in accidents of large-scale technological systems." *Process Safety and Environmental Protection* Vol. 97 (2015): pp. 13–24.

[8] Walker, Guy H, Stanton, Neville A, Kazi, Tara A, Salmon, Paul M and Jenkins, Daniel P. "Does advanced driver training improve situational awareness?" *Applied ergonomics* Vol. 40 No. 4 (2009): pp. 678–687.

[9] Broughton, Edward. "The Bhopal disaster and its aftermath: a review." *Environmental Health* Vol. 4 No. 1 (2005): p. 6.

[10] Salmon, Paul M, Walker, Guy H and Stanton, Neville A. "Pilot error versus sociotechnical systems failure: a distributed situation awareness analysis of Air France 447." *Theoretical Issues in Ergonomics Science* Vol. 17 No. 1 (2016): pp. 64–79.

[11] Woods, David D. "Four concepts for resilience and the implications for the future of resilience engineering." *Reliability Engineering & System Safety* Vol. 141 (2015): pp. 5–9.

[12] Dekker, Sidney, Hollnagel, Erik, Woods, David and Cook, Richard. "Resilience Engineering: New directions for measuring and maintaining safety in complex systems." *Lund University School of Aviation* Vol. 1 (2008): pp. 1–6.

[13] Roth, Emilie M, Multer, Jordan and Raslear, Thomas. "Shared situation awareness as a contributor to high reliability performance in railroad operations." *Organization Studies* Vol. 27 No. 7 (2006): pp. 967–987.

[14] Endsley, Mica R. "Toward a theory of situation awareness in dynamic systems." *Human factors* Vol. 37 No. 1 (1995): pp. 32–64.

[15] Endsley, MR and Jones, WM. "A Model of Inter-and Intrateam Situation Awareness: Implications for Design. New Trends in Cooperative Activities: Understanding System Dynamics in Complex Environments." *Human Factors and Ergonomics Society, CA* (2001).

[16] Stanton, Neville A, Stewart, Rebecca, Harris, Don, Houghton, Robert J, Baber, Chris, McMaster, Richard, Salmon, Paul, Hoyle, Geoff, Walker, Guy, Young, Mark S et al. "Distributed situation awareness in dynamic systems: theoretical development and application of an ergonomics methodology." *Ergonomics* Vol. 49 No. 12-13 (2006): pp. 1288–1311.

[17] Salmon, Paul M, Stanton, Neville A, Walker, Guy H and Jenkins, Daniel P. *Distributed situation awareness: Theory, measurement and application to teamwork*. CRC Press (2017).

[18] Salmon, Paul M and Plant, Katherine L. "Distributed situation awareness: From awareness in individuals and teams to the awareness of technologies, sociotechnical systems, and societies." *Applied Ergonomics* Vol. 98 (2022): p. 103599.

[19] Stanton, Neville A. "Distributed situation awareness." (2016).

[20] Kitchin, Joanne and Baber, Chris. "A comparison of shared and distributed situation awareness in teams through the use of agent-based modelling." *Theoretical Issues in Ergonomics Science* Vol. 17 No. 1 (2016): pp. 8–41.

[21] Nazir, Salman, Sorensen, Linda J, Overgård, Kjell Ivar and Manca, Davide. "How distributed situation awareness influences process safety." (2014).

[22] Stewart, Rebecca, Stanton, Neville A, Harris, Don, Baber, Chris, Salmon, Paul, Mock, Mel, Tatlock, Kerry, Wells, Linda and Kay, Alison. "Distributed situation awareness in an Airborne Warning and Control System: application of novel ergonomics methodology." *Cognition, Technology & Work* Vol. 10 No. 3 (2008): pp. 221–229.

[23] Fioratou, Evridiki, Flin, Rhona, Glavin, Ronnie and Patey, Rona. "Beyond monitoring: distributed situation awareness in anaesthesia." *British journal of anaesthesia* Vol. 105 No. 1 (2010): pp. 83–90.

[24] Alhaider, Abdulrahman A, Lau, Nathan, Davenport, Paul B, Morris, Melanie K and Tuck, Christopher. "Distributed Situation Awareness in Patient Flow Management: An Admission Case Study." *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, Vol. 62. 1: pp. 563–567. 2018. SAGE Publications Sage CA: Los Angeles, CA.

[25] Macquet, Anne-Claire and Stanton, Neville A. "Do the coach and athlete have the same «picture» of the situation? Distributed Situation Awareness in an elite sport context." *Applied ergonomics* Vol. 45 No. 3 (2014): pp. 724–733.

[26] Alhaider, Abdulrahman A and Lau, Nathan. "Resilience to the COVID-19 Pandemic: A Distributed Situation Awareness Perspective." *2020 Resilience Week (RWS)*: pp. 126–132. 2020. IEEE.

[27] Tan, James JY, Otto, Kevin N and Wood, Kristin L. "Relative impact of early versus late design decisions in systems development." *Design Science* Vol. 3 (2017).

[28] Leveson, Nancy, Daouk, Mirna, Dulac, Nicolas and Marais, Karen. "Applying STAMP in accident analysis." *NASA Conference Publication*: pp. 177–198. 2003. NASA; 1998.

[29] Zhang, Yingyu, Dong, Chuntong, Guo, Weiqun, Dai, Jiabao and Zhao, Ziming. "Systems theoretic accident model and process (STAMP): A literature review." *Safety science* Vol. 152 (2022): p. 105596.

[30] Ishimatsu, Takuto, Leveson, Nancy G, Thomas, John, Katahira, Masa, Miyamoto, Yuko and Nakao, Haruka. "Modeling and hazard analysis using STPA." (2010).

[31] Chatzimichailidou, Maria Mikela and Dokas, Ioannis M. "Introducing RiskSOAP to communicate the distributed situation awareness of a system about safety issues: an application to a robotic system." *Ergonomics* Vol. 59 No. 3 (2016): pp. 409–422.

[32] Hulse, Daniel E. "A Computational Framework for Resilience-Informed Design." (2020).

[33] Hulse, Daniel, Walsh, Hannah, Dong, Andy, Hoyle, Christopher, Tumer, Irem, Kulkarni, Chetan and Goebel, Kai. "fmdtools: A fault propagation toolkit for resilience assessment in early design." *International Journal of Prognostics and Health Management* Vol. 12 No. 3 (2021).

[34] Irshad, Lukman and Hulse, Daniel. "Can Resilience Assessments Inform Early Design Human Factors Decision-making?" *IFAC-PapersOnLine* Vol. 55 No. 29 (2022): pp. 61–66.

[35] Irshad, Lukman and Hulse, Daniel. "Resilience Modeling in Complex Engineered Systems With Human-Machine Interactions." *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Vol. 86212: p. V002T02A024. 2022. American Society of Mechanical Engineers.

[36] Artman, Henrik and Garbis, Christer. "Team communication and coordination as distributed cognition." *9th Conference of Cognitive Ergonomics*: pp. 151–156. 1998.

[37] Hutchins, Edwin. *Cognition in the Wild*. 1995, MIT press (1995).

[38] Stanton, Neville A Dr, Salmon, Paul Dr and Walker, Guy H Dr. *Systems thinking in practice: applications of the event analysis of systemic teamwork method*. CRC Press (2018).

[39] Kitchin, Joanne and Baber, Chris. "The dynamics of distributed situation awareness." *Proceedings of the human factors and ergonomics society annual meeting*, Vol. 61. 1: pp. 277–281. 2017. Sage Publications Sage CA: Los Angeles, CA.

[40] Stanton, Neville A, Plant, Katherine L, Revell, Kirsten MA, Griffin, Thomas GC, Moffat, Scott and Stanton, Maggie. "Distributed cognition in aviation operations: a gate-to-gate study with implications for distributed crewing." *Ergonomics* Vol. 62 No. 2 (2019): pp. 138–155.

[41] Stanton, Neville A and Harvey, Catherine. "Beyond human error taxonomies in assessment of risk in sociotechnical systems: a new paradigm with the EAST 'broken-links' approach." *Ergonomics* Vol. 60 No. 2 (2017): pp. 221–233.

[42] Yusuf, Sagir M and Baber, Chris. "Formalizing Distributed Situation Awareness in Multi-Agent Networks." *IEEE Transactions on Human-Machine Systems* Vol. 52 No. 6 (2022): pp. 1166–1175.

[43] Bruneau, Michel, Chang, Stephanie E, Eguchi, Ronald T, Lee, George C, O'Rourke, Thomas D, Reinhorn, Andrei M, Shinozuka, Masanobu, Tierney, Kathleen, Wallace, William A and Von Winterfeldt, Detlof. "A framework to quantitatively assess and enhance the seismic resilience of communities." *Earthquake spectra* Vol. 19 No. 4 (2003): pp. 733–752.

[44] Cabral, Sam. "Southwest and FedEx planes nearly collide at Texas airport." *BBC* Accessed 2023-03-09, URL https://www.bbc.com/news/world-us-canada-64541670.